だれにもわかるマイクロコンピュータ・プログラム

日本電気株式会社 電子デバイス販売事業部　マイクロコンピュータ販売部
東京都港区芝5丁目33-7(徳栄ビル) ☎(03)453-5511(大代)〒108

〈使うシステムをお求めの方に〉

●COMPO BS/80-A

238,000円

デスクサイドですぐに使えるマイコンシステムです。もちろん拡張性に対しても万全の備え。本格的なマイコンの実践応用に最適なシステムです。

オートカセット内蔵／NEC LEVEL-II BASICを内蔵／機械語でもプログラミング可能／万全の備え3電源／メモリボード、インターフェースボード、ユニバーサルボードなどの追加実装が可能

(ケースのみの販売もいたします。22,500円)

●パーソナルコンピュータ　PC-8001

NECの誇るLSI技術とICメモリの技術を結集。あくまでも高性能化を追求し、しかも最小システムの低価格化を実現しました。パーソナルコンピュータとして、ホビィはもちろんのこと業務用・研究用・教育用に…… 新しいコンピュータ時代を切り拓く、操作性に優れた期待のシステムです。

LEVEL-II BASICを一段と強化、演算精度は最大16桁／255文字と豊富な入力文字／プログラム領域を大幅に拡大する16/32Kバイトの大きなRAM容量／8色のカラーディスプレイ機能／操作性に優れたプログラマブル、ファンクションキーの採用／豊富な周辺機器

品　名
パーソナルコンピュータ(PC-8001)
拡張ユニット(PC-8011)
80桁ドットインパクトプリンタ(PC-8021)
40桁サーマルプリンタ
フロッピーディスクユニット(PC-8031)
12″カラーディスプレイ(高解像度)
12″カラーディスプレイ(標準)
12″グリーン・ディスプレイ

マイコンを志ざす方のための
マイコンの殿堂――ビット・イン

Bit-INN 東京 ☎(03)255-4575～6
〒101東京都千代田区外神田1-15-16ラジオ会館7F
●東日本地区通信販売店
日本電子販売株式会社☎(03)255-4571代
〒101東京都千代田区外神田1-16-1万世ビル3F

Bit-INN 大阪 ☎(06)647-2747～8
〒542大阪市南区難波新地6番町10-1
マスザキヤビル4・5F
●西日本地区通信販売店
ミカサ商事株式会社☎(06)942-1941代
〒540大阪市東区島町2-5

Bit-INN 名古屋 ☎(052)263-0971
〒460名古屋市中区大須4-11-5杏林殖産ビル2F
(地下鉄、上前津駅下車、万松寺方面へ)
●中部地区通信販売店
萩原電気株式会社☎(052)931-3511
〒461名古屋市東区東桜2-3-3

Bit-INN 横浜 ☎(045)314-7707～9
〒220横浜市西区北幸1-8-4横浜西口第2ミナトビル7F
(横浜駅西口より徒歩3分)
●通信販売店
日本マイクロコンピュータ株式会社☎(03)230-0041代
〒102東京都千代田区麹町4-5-21睦ビル7F

●NECマイコンショップ

下記のマイコンショップをBit-INN同様お気軽にご利用ください。

●北海道地区
大阪屋　☎(011)221-0181
〒060札幌市中央区北一条西3丁目大阪屋3F

●富山地区
インパルス　☎(0764)91-2212
〒930富山市五番町4-10西野ビル2F

●石川地区
北陸マイクロコンピュータ販売株式会社
☎(0762)21-3021
〒920金沢市此花町1-22中川ビル2F

●静岡地区
日興通信株式会社　☎(0542)55-7071
〒420静岡市伝馬町22-1小川ビル2F

●広島地区
インタフェース　☎(0822)49-3950
〒730広島市三川町10-10

●福岡地区
フルムラエルコン　☎(092)741-101C・751-6647
〒810福岡市中央区赤坂一丁目10-22

マイコンプログラム全集 1

——ラジオの製作別冊——

目　　次

第1章　プログラム作りの楽しみ

第2章　拡張プログラム集

第3章　応用プログラム集

マイコン対人間

モニタ画面活用

目　次

反射神経ゲーム

教育へのアプローチ

知的ゲーム

ラジオの製作別冊
マイコンプログラム全集　1
昭和54年10月30日発行

定価1,300円　(送料200円)

編集発行人　平 山 秀 雄
発 行 所　(株)電波新聞社
郵便番号141 東京都品川区東五反田１－11－15
電話(03) 445－6111(大代) 振替東京５－51961
印 刷 所　大日本印刷(株)
製 本 所　(株)堅　省　堂

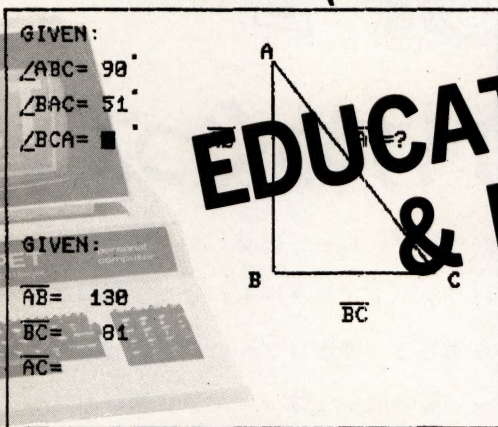
the progressive computer shop

コムスポット COM spot 共立

※コムスポット共立には、あなたの求めるソフトウェアが全て揃っています。業務用、教育・学習用、ゲーム用etc、あらゆる分野のソフト、アプリケーションプログラムの充実しているコムスポットソフトウェア・ライブラリーをぜひご利用下さい。

Software Library for Personal Computer

commodore PET, APPLE II, TRS-80, MZ80K

ベーシックマスター(MB6880), H68/TR&TV

AMORTIZATION SCHEDULE

PAYMENT	INTEREST	PRINCIPLE	BALANCE
100	16.66	83.33	916.666
100	15.27	84.72	831.944
100	13.86	86.13	745.810
100	12.43	87.56	658.240
100	10.97	89.02	569.211
100	9.486	90.51	478.697

INITIAL LOAN BALANCE? 1000
AMOUNT OF MONTHLY PAYMENT? 100
INTEREST RATE (PERCENTAGE)? 20

NEXT SIX MONTHS?

◆メディア(カセットテープ、ディスケットetc)も豊富に取り揃えております。◆

※マイコンを自由にさわれるコンピューターショップ※

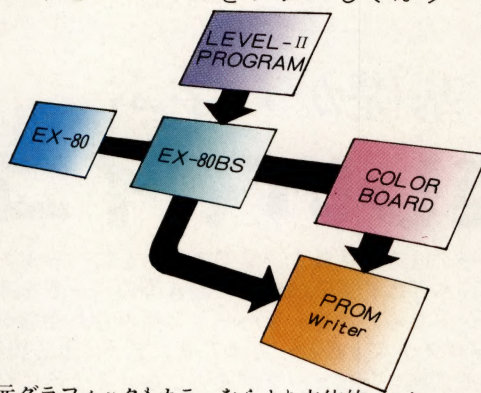
〒556 大阪市浪速区日本橋筋5丁目47日本橋会館2F
TEL (06)644-4666
5-47, NIHONBASHISUJI, NANIWA-KU, OSAKA, JAPAN.

TOSHIBA

——明日をつくる技術の東芝——

ドオシマスカ？

カラフル3次元。

●東芝のEX-80は、カラーボードとレベルIIをプラスしてカラーシステムにレベルアップ

たとえば迷路ゲーム。白黒ではさえない3次元グラフィックもカラーならより立体的。これにレベルIIをプラスすれば奥行の角度に変化をもたせることができ、表現力はさらに豊かになります。この、EX-80をレベルアップするのが新発売のEX-80BS用カラーボードとEX-80BS用LEVEL-II BASIC ROM。マイコンキットEX-80とEX-80BSと一体となりEX-80カラーシステムを構成します。

〈新発売〉EX-80BS用カラーボード

EX-80およびEX-80BSの2枚と組合せ、カラー表示プログラムをベーシック言語で簡単に作成。色は赤・緑・青の3原色をベースに、8種類(白・黄・シアン・緑・マゼンタ・赤・青・黒)のカラー表示ができ、TVゲームなどがカラーで楽しめます。

★放電プリンタやフロッピーディスクなどの入出力機器が接続できるよう3個のI/Oポートが用意されています。

★ビデオRAM最大6Kバイト、ユーザエリアとしてROMエリアが最大8Kバイト用意されています。

★PROMライタボードが接続できます。(EX-80BSにも接続可)

〈新発売〉EX-80BS用LEVEL-II BASIC ROM

マスクROM3個をBSボードに実装。

★実数演算ができるため、例えばカラーボードと組合せることにより放物線などが描け、処理の高級化が図れます。

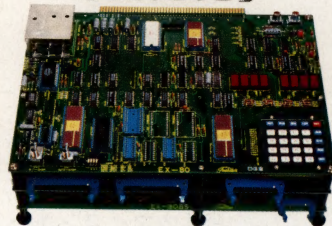
★浮動少数点演算ステートメント(加・減・乗・除算など有効ケタ数を9桁に拡張)

★三角関数・対数・指数関数・ルート・初等関数など高度な組込関数を用意しています。

★カラー表示ステートメントが完備しています。

〈新発売〉EX-80BS・EX-BSカラーボード用PROMライタ

PROM TMM323C専用のライタ。16Kビット(2K×8ビット)単位のデータ保存が可能になりました。

[新発売]

お問合せは… **東芝マイコンセブン**

〒101東京都千代田区外神田3-13-7ニューカクタX-1ビル5F
TEL(03)255-7588・9〈10:00AM~6:00PM,水曜・木曜定休〉

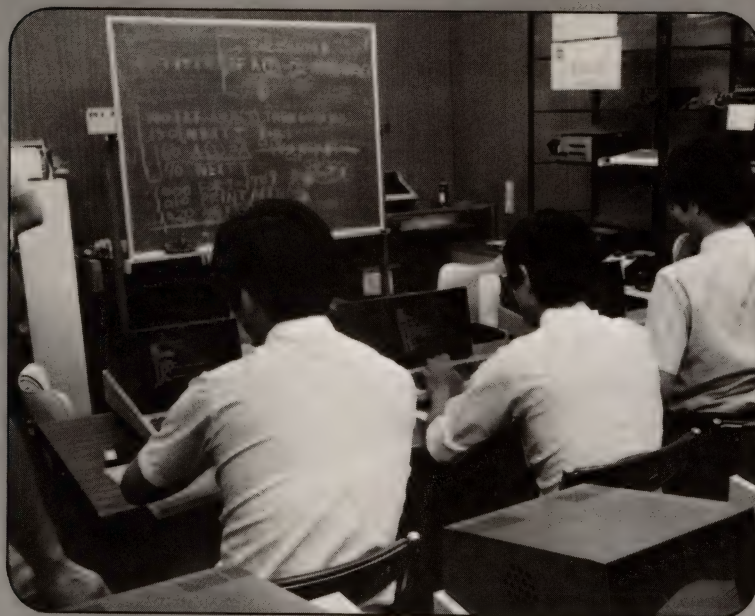
東芝マイクロコンピュータキット

EX-80 LEVEL-IIバージョン

東京芝浦電気株式会社　半導体営業推進部
マイクロコンピュータ課
〒210川崎市幸区堀川町72
TEL(044)522-2111(大代)

東芝

第1章

エンジョイ
マイコン・プログラミング!

豊かな生活を創る！

エンジョイ・マイコンプログラミング

福島憲一

この世にコンピュータと呼ばれる物が誕生してから30余年がたちました。最初のコンピュータと言われているENIACは，床面積1000㎡を要し重量30トン，1万8千本の真空管から発せられる熱量はへたなビルの暖房装置以上だったそうです。学生時代，私の大学にも国産の真空管式コンピュータが博物館的な意味で置いてありました。働らかせようとすると，日に何本かは真空管を交換せねばならず交換するとフリップフロップのバランス調整に又時間がかかるという大変な代物でした。今ではLSI化されたCPU1個がこれらの性能をはるかに上廻っています。

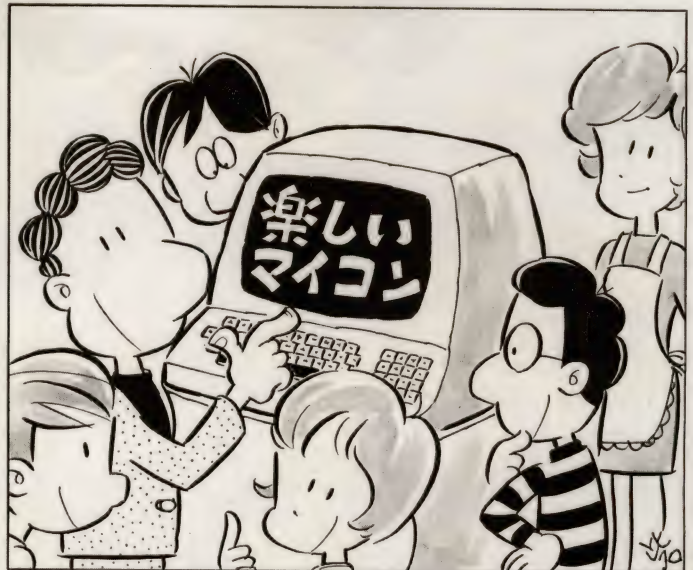
一方，同様にメモリも高速化小型化され、消費電力も減り量産されて大幅にコストダウンした事により，コンピュータは計算機から情報処理装置へとイメージを変えてあらゆる分野に利用されるようになりました。同時に1台のマシンをたらい廻し的に多目的に使う状態から専用化された分散処理へと利用形態も変ってきました。それに伴い，コンピュータ言語も多様にわたる事になり，そして，ホームコンピュータ，ホームプログラマの時代へ入ってきました。

私達は幸せです。「自宅にコンピュータを」なんて夢の又夢だったのですから。貴男がお持ちのマイコンにしたところで20年前に同等の性能を持った物を買おうとしたら、大卒初任給500カ月分（42年分？停年を過ぎてしまう！）程が必要だったでしょう。せいぜいマイコンと仲良くしてその性能をフルにひき出してやらねばバチが当たるというものです。

自分のマシンをもっと理解しよう

“敵を知り己を知れば百戦危うからず”という言葉があります。又，管理職は部下の長所短所を知り尽くしてこそその才能を生かせるでしょう。

コンピュータも同じ事。マニュアルを読んだだけで判った気になってはいけません。マニュアルに書いてない事に疑問を持ち，どんどん実行させてみるのです。そして、どういう場合にちゃんと実行される，或はエラーが出るか，実行後の変数内容はどうなるか等を自分で調べるのです。

機械語に自信の有る方はモニタやインタプリタの内容まで分析するのが理想的ですが，まあそこまでやらずともBASICの動作については色々やってみるべきです。たとえダンプに積む程エラーを続出させたとしても，それが原因でマシンが壊れたり絶対しません。エラーを出せば出す程自分

のBASICに対する理解は深まっていると思いましょう。“もっと，エラーを！！”

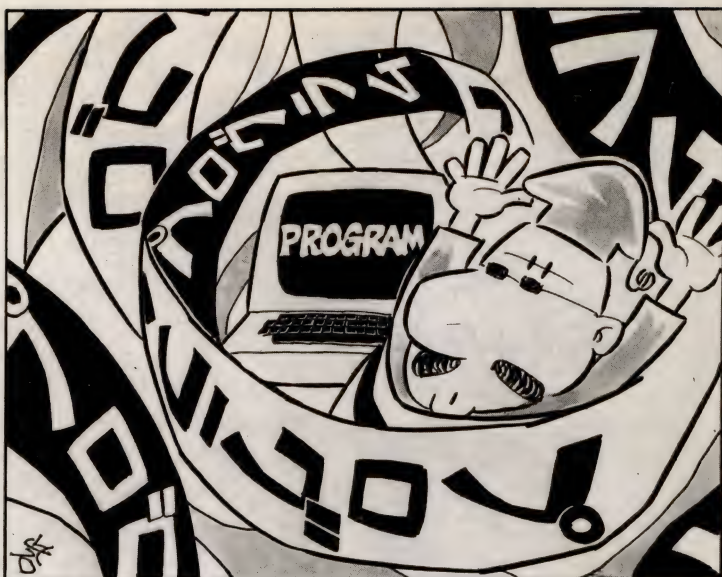
名人は段取り7分

何かプログラムを作ろうと思いたったら，どの様に進めれば良いでしょう。

まず机に向かい悠然と煙草などふかし，良いウイスキーかブランデー等をストレートでチビリチビリやりながら，焦点の定まらぬ眼を中空に向けて瞑想にふけります（手ぶらだとアホみたいに見えるから気を付けましょう)。やがておもむろにフローチャートを書き始め，終わったらそれを眺めて女房が文句言おうが子供が騒ごうが再び沈思黙考。「よし！」とかけ声をかけると愛機に火を入れ入力を行い，1発で動いたプログラムに長嶋監督の様な眼をして「ムヒヒ……」と昏の端で笑う。これが理想的でしょう（但し，離婚されても責任持ちません)。

一方，思いつくやすぐに，キー・ボードを叩き始める人がいます。途中で遊びに出たりテレビを見ては「エート，どこまで進んだんだったかな」なんて言いながら再開します。

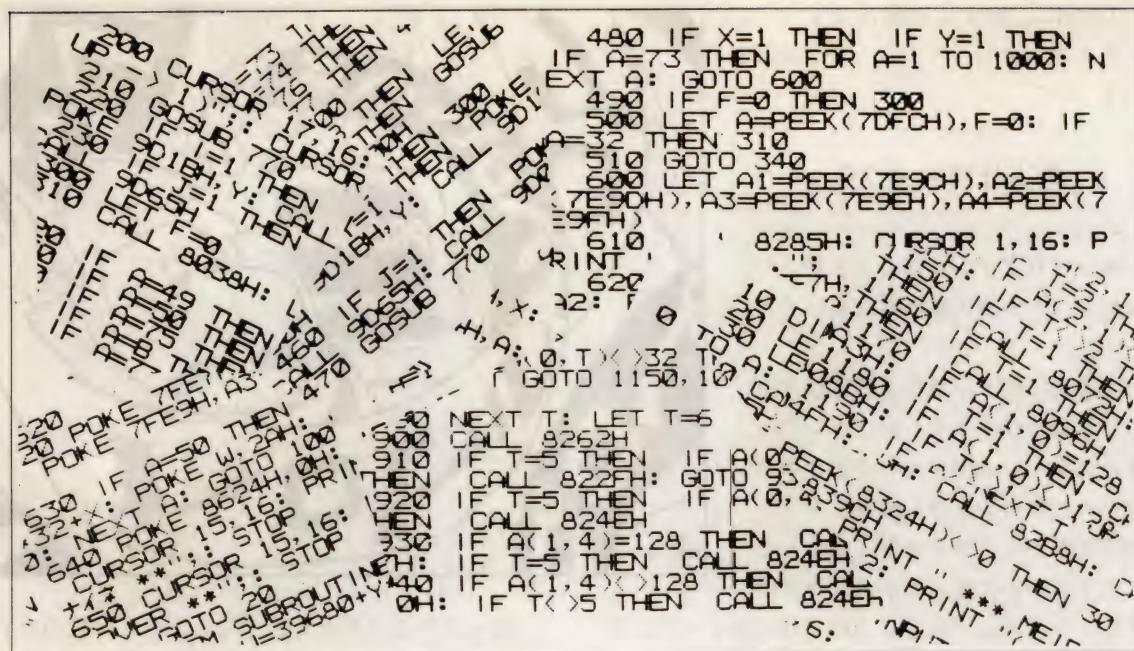
RUNさせてエラーの続出にも「あー，ソウカソウカ」等と言いながら全然へこたれません。やがて必然性の無い変数やGŌTO文の大安売となった頃プログラムは走り出し「ヤッタ！」と飛び上って喜びます。前者を眠狂四郎流円月殺法型とすれば後者は，ブルースリー流腕力型とでも申しましょうか。大工さんで言うなら前者は午前中は材料の寸法を測ったり板目を見たり道具を磨いては並べているだけ。しかし仕事にとりかかるや，動きに無駄が無く組上りはピタリというタイプ。

後者はいきなり材料を切り始めるのだが，寸法が違って切りなおしたり開きすぎた狭間に端切を叩き込んでゴマかし，「あれ，トンカチはどこへ置いたかな」等とウロウロ捜したり，威勢の良い割には仕事のはかどらないタイプ。そしてどちらが仕上りが美しく増改築時にも問題が無いかは明らかでしょう。

植木屋さんでも大工さんでも，名人は段取り7分。プログラム作りも全く同じで，アルゴリズムやパターンの分析が終わって正確なフローチャートが書ければ，仕事は終わった様な物です。大きなプログラムを作る為にも，又，保存性の点からもフローチャートを書くのは絶対の原則です。

断じて前者のスタイルで行くべきです。がー，しかし，後者も仲々面白いですから，たまにはやりましょう（編集部註：エライ奴に原稿頼んだなあ)。とにかく良いプログラムを作るには与えられたテーマを分析して解決する力がなくてはなりません。その為には幅広い知識を身に付ける事。数多くこなす事。人の作ったプログラムに学ぶ点があればどんどん取り入れる事です。

この本も単に自分のマシンに合った頁だけ意味も判らず入力するのなら勿体ないのです。買うの止めましょう（編集部註：コラッ担当者！この筆者ナントカしろ！！)。しかし，目的に対して各作者がいかにアプローチしているかを理解し，勉強材料として或は批判材料として有効利用されるなら大いに価値があります。やっぱり立読みだけでなく買いましょう（編集部註：フーッ，ヤレヤレ！)。

BASICの書き換え

世はマイコン花咲かりで，色んな機種が発売されています。そしてそのほとんどがキー・ボード

にCRTディスプレイ，カセット，言語はBASICという装備です。しかしながら異機種間でソフトの完全な互換性が得られるものは有りません。

その理由はハードに起因する物とソフトに起因する物とがあります。まずハードによる差ですがこれは改造しない限り，できる機種はできるができない機種はできないであきらめていただくしかありません。例えば一部の機種が音を出す為のI/Oを持ち，BASICのMUSICコマンドでアクセスできるところを他機種は無視する以外ありません。次に肝心のCRTディスプレイですが表示可能なキャラクタ数が，32×16，64×16，40×25等々まちまちなのです。更に加えてキャラジェネのパターンの違いです。これらはPRINT文の内容を縮小するなり，キャラクタコードの数値を変えるなり，ソフトの方でなんとか解決せざるを得ません。

さて，ソフトによる違いです。最近ではBASICも8K以上の物が主流になって，コマンドや関数等も多くなり機能強化されていますが，この本には4K以下のいわゆるLEVEL-1と称される物も含まれています。同じBASICでもインタプリタの大きさ，設計思想の違い等により仕様はかなり異なる物で，良く言えば各社個性を出していると言えます。ハードと違ってソフトの差異は何とか対処の方法があるもんですが，自分で機械語のサブルーチンを作ってBASICとリンクさせる以外手が無い様な場合も有ります。

例えば

1．整数演算型BASICで浮動小数点演算をしたい場合

2．プログラム中にカセットでデータを読み書きしたいが，適するステートメントが無い場合

これらあたりが少々やっかいでしょう。

他に御存知とは思いますが，代表的な書き換えの例を掲げておきます。

1．DEFFW（関数定義）が無い場合

何か変数を設定して計算式のサブルーチンを作りGOSUBで処理する。

2．1文字入力（GET等）が無い場合

キー・ボードはメモリマップドI/Oになっている筈ですから，そのアドレスをPEEKして処理します。

3．ON GOTOが無い場合

GOTO（計算式）で処理する

4．SET, RESET等画面全体をグラフィック的に扱うステートメントが無い場合

グラフィックに適するキャラクタがあればサブルーチンを作ってしまう方が楽です。

5．配列が一次元しか無い

配列添字を式にして処理する

6．変数がA～Zだけで足りない

配列を変数として使います

7．ストリング変数を持たないBASICでストリング処理（VAL，LEN等）したい

必要なメモリ領域を確保してPEEK，POKEで処理する（ちょっと面倒，機械語のサブルーチンを作った方が楽かも）

他に重要な事として画面の希望の位置に表示させる方法が大別して3通りあります。

1．CURSOR X，Yを使う物
2．カーソル移動マークをストリングとして使う
3．PRINT ⓐを使う物

又，POKEで表示させている場合は当然V-RAMのアドレスに合わせて変更しなければなりません。まあこれらはRUNさせて画面を見て修正できるから楽でしょう。お持ちの機種以外のV-RAMアドレスはあまり御存知ないでしょうから，参考資料として掲げておきます。（ ）内は，10進数で表現した場合です。

TK-80BS	7E00～7FFFH	(32256～32767)
MZ-80K	D000～D3E7H	(53248～54247)
PET	8000～83E7H	(32768～33767)
MB-6880	0100～03FF	(256～1024)
TRS-80	3C00～3FFFH	(15360～16384)

現在の機器構成ではCRTに出力する事が最終目的になるので，プログラムの実行につれてカーソル位置がどの様に動くかフォローしてプログラムする必要が有ります。もう一つの注意事項。

例えば次の様なプログラム

```
10  CLEAR : PRINT "    MAN IS STRONG"
20  CURSOR1, 1: PRINT "WO"
30  END
```

CLEARとCURSOR 1，1の部分は機種によって表現が異なるでしょうが，どの機種もこのプログラム実行後カーソルは2行目の先頭にある筈です。しかしCRT上に残るプログラムの実行結果は，WOMAN IS STRONGと残る機種とWOだけしな残らない機種とがあります。つまりPRINT文を実行後次行の先頭にカーソルを移動させる際，デスラ戦法風にワープして行く機種と，行の残りをごていねいにクリアして通り過ぎて行く機種が有るのです。せっかくのゲーム等も表示結果がまるで違ってきます。その他コンマとセミコロンの使い方等細かい違いも有りますので注意して書きなおして下さい。

ちょっと雑談

過日友人達と飲む機会が有りました。A氏はソフト専門の人，B氏は周辺機器関係のハード，ソフトを主としてやっています。C氏は石油会社に勤めコンピュータと縁の無い人。ひとしきり石油問題でC氏を袋叩きにした後「プログラミングとは？」と禅問答みたいな問いかけにA氏が答えました。「小説を書く様なもんだ」B氏も答えました。「アホな奴に何か教える様なもんだ」ナル程，どちらも言えてますなあ。皆さんはどちらの感覚に近いでしょう。

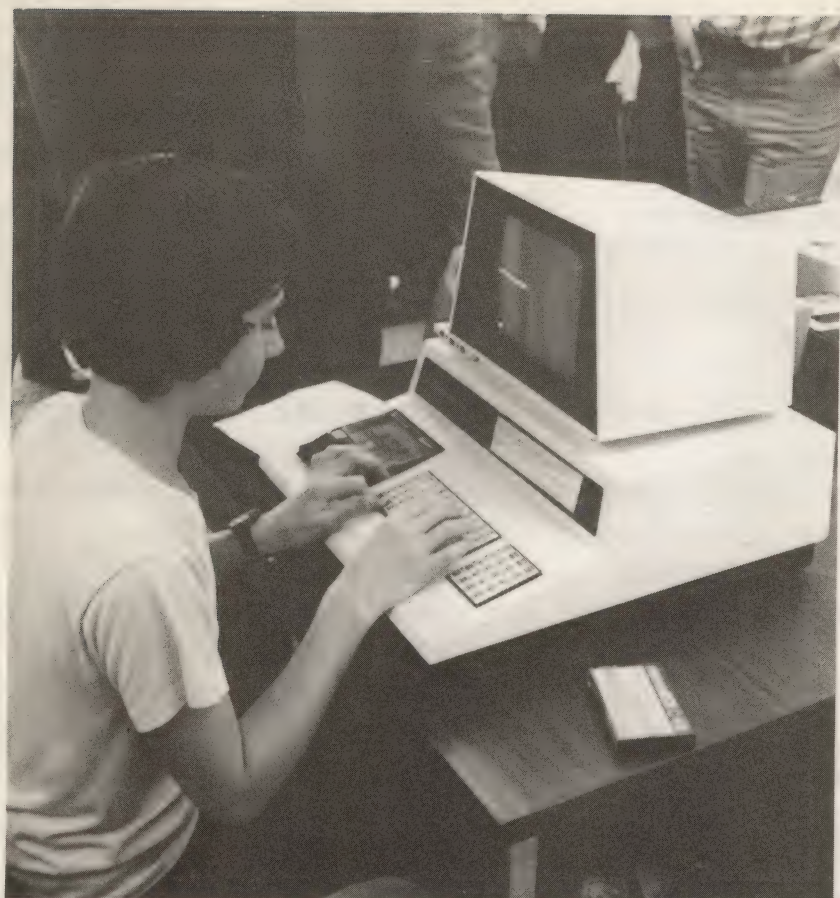
BASICが発表されたのは1964年（東京オリンピックの年）ですが，まさに今日のマイコン時代あるを見込した様な素晴しい言語だと思います。悪く言う人も居ますがBASIC無しでマイコンはこうも早くワンボードから進化し得たでしょうか。コンピュータに縁の無い人達をこの世界に導けたでしょうか。今BASICでマイコンと接している皆さんがどちらの感覚に近いか大いに興味のあるところです。

さて，その後しばしコンピュータ談議に花が咲いたのですが，ふと見るとC君が話に加われず淋しそうにしています。結局しぶるC君にマイコンをやらせ教育しようという事に衆議一決（といっても3人だけど）しました。それも授業料は各自にボトル1本という格安で(何と優しい友人達でしょう！)。1か月程私のマイコンを借りていったC君は興味が湧いてきたのかやがて愛機を求め今では機械語にも挑戦しており，コンピュータコンプレックスはどこへやら,「難かしいプログラム考えてる最中に電話するな」と怒られる時もあります。ガンバレ！C君！　皆さんもなるべく多くの人にプログラミングの楽しさを教えてあげて欲しいのです。お願いします！

トレーニングモジュールH68 TRシステム
スタテックメ

第2章

プログラム開発
実用アイデア集

1

LEVEL-Iプログラム復活！

ワンタッチ LEVEL-Ⅰ ⇄ LEVELⅡ 切替えシステム

TK-80BS

竹下　洋

TK80BSは，はじめ，BASICとしてLEVEL-Ⅰを装備して販売されました。その後LEVEL-Ⅱ BASICのROMが無料配布され一段とマイコンとしての機能がアップしたわけですが，その結果これまでLEVEL-Ⅰで組み，貯わえてきたプログラムは全てLEVEL-Ⅱの文法に合う様に変更し，再度キーボードから入力するか，LEVEL-ⅡとLEVEL-ⅠのROMを差し換えなければ走らなくなりました。

しかしこの様な面倒な事をいちいちやってはいられません。

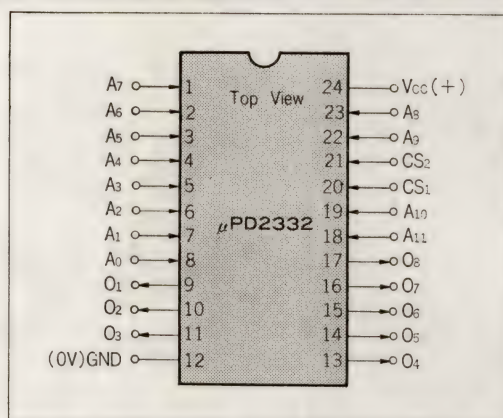
そこで，BSを大改造することなしにこれらを変換できるように改造します。LEVEL-ⅠのROMを持っている人はぜひ改造してみてください。

LEVEL-Ⅰ，LEVEL-Ⅱ 変換の原理

ロータリ・スイッチなどでLEVEL-ⅠのROMとLEVEL-ⅡのROMのピンを全部切替えてもよいのですが，これではあまりにも大変です。

BASICインタプリタの書き込まれている32KビットROM，μPD2332の端子図，**第1—1図**を見て下さい。

V_{CC}，GND，アドレス（A_0～A_{11}）はそれぞれ並列に継げばよさそうです。

さて，残ったのはデータ（D_1～D_8）端子8本とチップセレクト（C_S）端子2本です。

《第1—1図》μPD2332の端子

データ端子は，データバスに継いだらどうなるでしょう？　LEVEL-ⅠとLEVEL-Ⅱのデータがごちゃごちゃになるかな，いやだいじょうぶです。

LEVEL-Ⅰ，LEVEL-Ⅱのどちらか一方のROMしかチップセレクトさせなければ，一方のROMのデータ以外はデータバスに乗らないのです。

そうするとチップセレクト以外は全て並列に継なげばよいのですから，何とチップセレクト（C_S）端子2本だけ切替えればよい事になります。

BSの基板でインタプリタを走らせてチップセレクトの信号を調べてみたらC_{S2}は常時5Vなのです。有難い!!

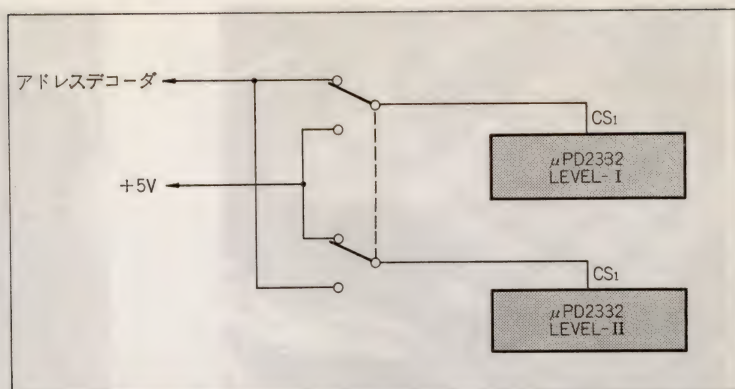
さて，いよいよ最後のC_{S1}ですが，CPUから出されたアドレス信号がE＊＊＊Hの時アドレスデコーダからこのROMのC_{S1}にLの信号を送り込んで

チップセレクトしています。

従って LEVEL-Ⅰ の ROM と LEVEL-ⅡのROMをC_{S1}を除いて全て並列に接続し，LEVEL-Ⅰを走らせたい時はそのROMのC_{S1}にアドレスデコーダからの信号を送り，LEVEL-Ⅱ の ROMのC_{S1}をHに引き上げておけばよいことになります。

もちろんLEVEL-Ⅱを走らせる場合は，その逆にすればよいのです。**第1－2図**に回路図を書いておきました。

《第1－2図》LEVEL Ⅰ，Ⅱ切替えの回路

《第1－3図》BS基板面の配線図

実際の配線

○基板の改造

配線図を**第1－3図**に示します。BSの基板を裏にし，μPD 2332Cが3個並んでいる所を捜します。

C_{S1}以外は3個共並列につながっているはずです。

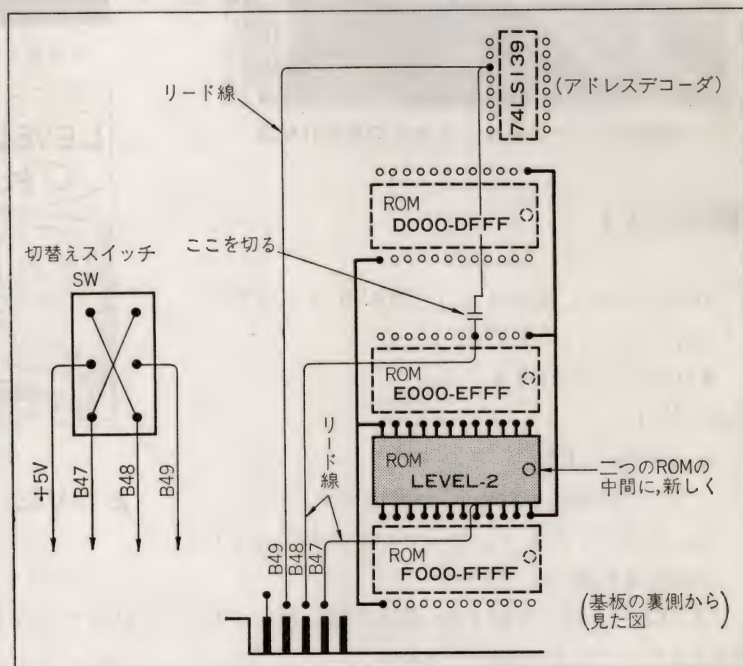
真中のROMがBASICインタプリタ（E000～EFFF）ですから，このROMのC_{S1}を見つけ，アドレスデコーダ74LS139Nからきている線をNTカッター等でカットします。

今カットしたC_{S1}の端子及びアドレスデコーダの端子に各々ビニール線をハンダ付けし，他端をTK-80BSバスの空きB-48，B-49にハンダ付けします。

○ROMセケットの取付け

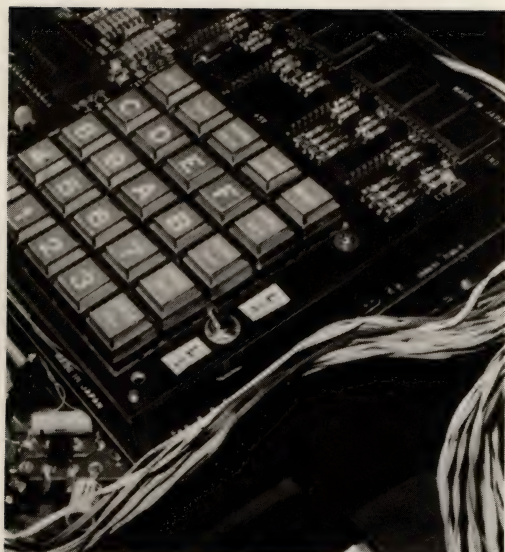
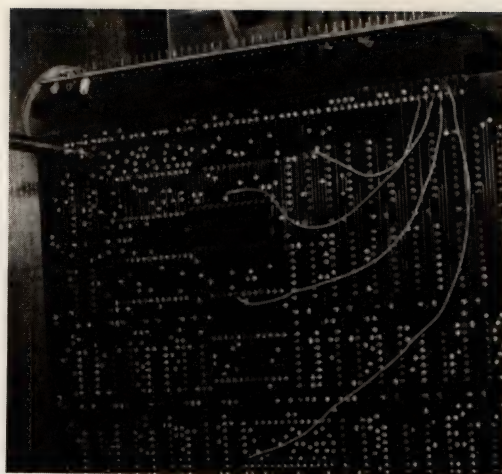
C_{S1}にあたるピンを外側に曲げ，ビニール線をハンダ付けし，他のピンに接触しない様チューブ等を挿入します。

BS基板を裏にし，二つのROM（E000～，F000～）の中間に方向を揃えておきます。

ピッタリと足が対応しているはずです。C_{S1}を除いて対応するピンをハンダ付けします。あらかじめつないでおいたC_{S1}のビニール線をTK-80BSバスの空きB-47にハンダ付けします。

○スイッチの取付け

写真を参考にして，TK-80にスイッチを取付けて下さい。LEDの下にある抵抗R_{18}のすぐわきに5Vがきています。これをスイッチの一方の中間点に接続します。

他方の中間点には，B-49を通してきたアドレスデコーダの出力を接続します。

残りのスイッチ端子をたすきに接続し，各々B-47，B-48に接続します。

《写真1—1》切替えスイッチの取付け位置

《写真1—2》BS基板裏側の配線のようす

動作テスト

ROMを実装し電源を入れてBASICインタプリタを呼び出すと，LEVEL-Ⅰなら

＊BASIC READY＊

LEVEL-Ⅱなら

＊LEVEL-2 BASIC V1.0＊

と，テレビ画面に表示されればOKです。

もし暴走するようでしたらすぐ電源を切ってもう一度配線を調べて下さい。

LEVEL-Ⅰ, Ⅱの切替えは，電源を切らなくてもRSTをかけて行えば暴走しません。

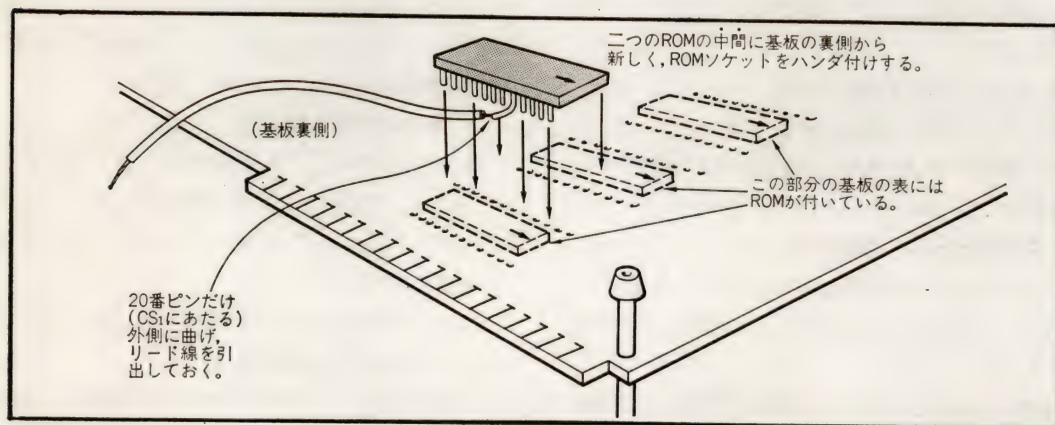
おわりに

これでLEVEL-ⅠまたはⅡで組んだプログラムは全て走らすことができます。

スイッチをLEVEL-Ⅰに倒し，これまで隅の方に置去りにされていたプログラムを走らせ，LEVEL-Ⅱでより高度なプログラムを開発して下さい。

《第1-4図》プリント基板裏側の工作

2

ディスアセンブラ機能付き

LEVEL2 ASSEMBLER

TK-80BS

鈴木俊孝

このTK80-BS用LEVEL-2　ASSEMBLERは機能がアップして，非常に使いやすいものとなっています。

では，その使い方などについて順に説明していきます。

使い方

プログラムを9000番地からRUNさせると,画面には＊LEVEL-2　ASSEMBLER　V1.0＊と出て，¥マークが出て入力待ちとなります。

コマンドには11種類あって，それは**第2－1表**のとおりです。それらのコマンドについて説明していきます。

〔1〕ORG（ORIGIN）

ORGは，ORG,××××のようにパラメータをつけて使います。ORG, 8C00と入力すると，画面に8C00_と表示され、入力持ちになります。ここでは3種類の入力方法があります。

(1)　インストラクションの入力

第2—2—1表を見てください。①は1バイト命令で，これは普通に入力します。②～⑤は2バイト命令で，2バイト命令には4種類の入力方法があります。②は一般的ですが,③のように最後にHをつけないと十進数とみなされます。④の“K”はASCIIコードのK, すなわち4BHです。⑤の“M”はJISコードのM，すなわち0DHです。⑥～⑧は3バイト命令で，これを見ればわかると思います。ただし⑥の最後にはHをつけることが必要で，つけないと3C19というラベルとみなされます。また，⑧の$+のあとは，十進法です（65535まで)。

(2)　データの入力

第2—2—2表を見てください。①は数値データの入力です。初めに必ず＋をつけます。②はASCIIコードの文字の入力です。この例では，41，53,

ORG	HDS
DSA	DEL
CON	NAL
LAB	NEW
PRG	MOV
CLR	

《第2—1表》

```
①POP  B
②ANI  3FH
③MVI  A, 16
④MVI  B, "K"
⑤CPI'M'
⑥JMP  3C19H
⑦CALL  PRINT
⑧LXI  D, $+7
```

《第2—2—1表》

```
①+2C,30,5,0,ED
② "ASCII"
③'JIS'
```

《第2—2—2表》

```
*LEVEL-2 ASSEMBLER V1.0*
¥NEW
¥ORG,8300
8300 _
```

《写真2—1》アセンブラ・イニシャル表示

```
*LEVEL-2 ASSEMBLER V1.0*
¥NEW
¥ORG,8300
8300 CLEAR
--LABEL INPUT
8300 _
```

《写真2—2》ラベル・インプット

《写真2—3》プログラミング完了

《写真2—4》¥表示モードにぬけだす

43, 49, 49とメモリに入ります。③はJISコードで，この例では0A, 09, 13とメモリに入ります。

(3) ラベルの入力

そのままPRINT［復改］と入力すれば，その番地にPRINTというラベルがつきます。また，削除する場合は,／（カタカナの「メ」のところ),［復改］と入力すればその番地のラベルが削除されます。そして，入力のときは——LABEL INPUT, 削除のときは——LABEL DELETEと表示されます。ここで注意しなければならないことが一つあります。すなわち，インストラクションとまちがえるようなラベルは入力できません。たとえば, STARTというラベルを入れようとすると，START と解釈されてしまいます。しかし絶対に入れられないわけではありません。ある番地でJMP START等として，あとで説明するNALコマンドで番地を入れればよいのです。

アセンブルをやめたいときは，すぐ［復改］と入力すればやめられます。また,［空白］［復改］と入力すると，その1行が逆アセンブルされ，そして次の行へいきます。

入力するときは、空白をいくつあけてもかまいません（ただし，インストラクション，ラベル等の間はあけてはいけません)。

プログラムに使える領域は，8C00～8FFFと8000～83C0です。

〔2〕DSA（DISASSEMBLER)

DSAは，DSA［復改］と入力する方法と，DSA,××××［復改］と入力する方法とがあります。前者は，あとで説明するプログラム領域の先頭から逆アセンブルするということです。また，プログラムが入っていない場合は，**ニュウリョク　アヤマリ**となります。

まずそのように入力すると，画面には15行逆アセンブルされた状態で，カーソルは最下行の左端にとまっています。この状態で使えるコマンドは**第2—3表**のとおりです。

N：Nを押すとつづく15行が逆アセンブルされます。

S：Sを押すと連続的にどんどん逆アセンブルさ

```
*LEVEL-2 ASSEMBLER V1.0*
¥DSA_
```

《写真2—5》ディスアセンブラ命令インプット

《写真2—6》ディスセンブラリスト

《写真2—7》ラベル（LAB）命令結果

《写真2—8》PRG命令

れます。とめるときは，他のキーを押します。

L：Lを押すとつづく1行が逆アセンブルされます。(ただし画面の下の方があいているときはいっぱいになるまでです。)

R：Rを押すと前の15行が逆アセンブルされます。（この場合は，前のアドレスをサーチするため，ほんのわずかですが時間がかかります。)

まっ消：まっ消を押すと最下行が1行消えます。いちばん上まで消してしまうと，自動的に¥マークが出て入力待ちになります。

復改：復改を押すと¥マークが出て入力待ちの状態になります（逆アセンブルをやめるとき)。

空白：空白を押すとカーソルのあったところに ? と出て，カーソルが一つ右に移動して入力待ちになります。このとき，16進4桁以内で番地を入力すれば，その番地から逆アセンブルされます。

0，1，2，……，F：n (0～Fのどれか) を押すとn000番地から逆アセンブルされます。

アドレスのついていないラベルがあったときは，機械語のところには―――――と出ます。たとえば，PRINTというラベルにアドレスがついていなければ，機械語部分にCD――――と出ます。

N	S
L	R
まっ消	
復 改	
空 白（スペース）	
0，1，2，…，F	

《第2—3表》

〔3〕CON（CONTINUE）

あるコマンドを実行して途中でやめた時等にそのつづきを実行するのに使います。ただしCONが使えるのはORG，DSA，LAB，NALの後だけです。

〔4〕LAB（LABEL）

LABと入力すると，**第2—1図**のようにラベルが全部表示されます。これは，ラベルが少ないときに画面に入りきるのですが，多いときには先に出たのがどんどん消えていってしまいます。これでは困るので，次の機能をつけてあります。

①ラベルがどんどん出ているときに，スペース・バーを押すと一時停止します。つづきを見ると

《写真2―9》8200番地のMOV命令

《写真2―10》移動した結果

《写真2―11》LAB，PRGによる再確認

《写真2―12》CRT，PAINTの実行

《第2―1図》ラベル表示機能

《第2―2図》アドレスの決っていないラベル表示

《第2―3図》プログラム領域の表示

きには，他のキーを押します。

②ラベルがどんどん出ているときに「！」を押すと，途中で表示が終わります。つづきを見たいときは，CONコマンドを実行します。

また，アドレスがきまっていないラベルのときは，**第2―2図**のように表示されます。

〔5〕PRG（PROGRAM）

PRGは，PRG［復改］と入力する場合と，PRG ,××××,××××と入力する場合とがあります。前者のときは**第2―3図**のようにプログラム領域が表示されます。また，NEWコマンドを実行したあとは，NO PROGRAMと出ます。後者はプログラム領域の定義です。これは，後で説明するMOVコマンドのときに使います。ORGコマンドによってプログラムを入力していくと，それにしたがって自動的にプログラム領域も広がっていきますから，普通は後者の使い方はしません。

〔6〕CLR（CLEAR）

BASICのCLEARと同じです。

〔7〕HDS（HARDCOPY DISASSEMBLER）

マイコンを持っている人の中には，プリンターを持っている人も大勢いると思います。プリンタ

《写真2—13》再確認

《写真2—14》?マークに変更

《写真2—15》ブレークキーでGO，8200

《写真2—16》一面?マークに変更

ーを持っている人たちのために特別に作ったコマンドがこのHDSです。

HDSはHDS, ××××，□□□□というように入力すれば，××××から□□□□までが逆アセンブルされてプリンターに出力されます。出力はBASICと同じく，一文字ごとにプログラムの実行が84ECにでてきます。BASICでHLSTを使っている人は，それと同じにやればいいのです。

〔8〕DEL（DELETE）

DEL, ××××, ××××という形で入力して，プログラムの一部を削除するのに使います。このとき，その部分は単に消えるのではなく，**第2—4図**のようにそれ以降END ADDRESSまでが前の方に移動します。このとき，PRGコマンドでみられるプログラム領域も自動的に縮まります。また，前の方に移動してくるのも，単に移動するのではなく，ちゃんと3バイト命令の2，3バイト目も変化します。それから，移動するところにあったラベルも自動的に移動しますが，削除した部分のラベルは消えませんので注意して下さい（先に削除する部分のラベルを消してからこのコマンドを実行して下さい）。

《第2—4図》DEL（削除）命令

〔9〕NAL（NO ADDRESS OP LABEL）

ラベルのうちでアドレスのついていないものに，アドレスを入力するのに使います。たとえばラベ

```
PRINT?
```

《第2—5図》NAL入力

```
PRINT?FA52
CLEAR?_
```

《第2—6図》NAL再入力

```
PRINT?
CLEAR?_
```

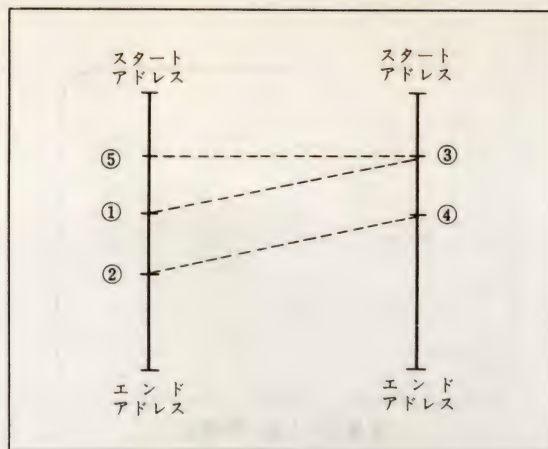
《第2—7図》CLEARの入力

ルがたくさんある中で，PRINTとCLEARというラベルにアドレスがついていなかったとすると，NALと入力すると画面は**第2—5図**のようになります。このとき16進4桁以内でアドレスを入力します。また入力するとこんどは**第2—6図**のようになりますが，たとえばCLEARだけ入力したい場合は，PRINT?とでたときに［空白］［復改］と入力します。すると画面は**第2—7図**のようになります。それから，もっとたくさんあったりして途中でやめたいときには，直接［復改］を入力します。再び始めるときはCONコマンドが使えます。

〔10〕NEW（NEW）

プログラムを入れる前には，ラベルにでたらめなものがたくさん，そしてプログラム領域もでたらめに入っていますから，プログラムを入れる前には必ずNEWコマンドを実行します。そうすると，ラベルは全部なくなり，PRGコマンドを入力するとNO PROGRAMと出るようになります。また別のプログラムを入れるときにもNEWコマンドを実行します（このへんはBASICと似ています）。

〔11〕MOV（MOVE）

これはプログラムの一部分を別の場所に移動するのに使います。これはモニタのMMというコマンドに似ていると思えばいいのです。しかし，モニタのMMコマンドはメモリの内容がただ移動するだけでしたが，MOVコマンドでは3バイト命令の2，3バイト目も変化し，ラベルの位置も自動的

《第2—8図》MOV（MOVE）命令

に移動します。また，プログラム領域もそれにしたがって変化します（しかしMOVコマンドを実行する前とした後には必ずプログラム領域をたしかめて下さい）。

たとえば**第2—8図**のように移動したいときは MOV，①，②，③と入力します。このとき，①と⑤の間のラベルは消えませんから注意して下さい。

また，このコマンドでプログラム全体を別の場所へ移動することも可能です。その場合は，プログラム領域もちゃんと移動します。

使い方の例外

コマンドを実行するほかに，直接入力する方法もあります。それは，たとえば84ECからJMP 8200Hというものだけを入れたい，というときにはORGコマンドをいちいち使っていてはたいへんですから，¥のあとに8200：JMP　8200Hと入力すればいいのです。この場合，8200：を自分で入力するということだけで，あとはORGコマンドと全く同じです。すなわち，ラベルもFA52：PRINTのようにして入れられるということです。

実際にやってみれば気がつくと思いますが，アドレスのついていないラベルは消せないことになっています。ですから，アドレスのついていないラベルを消すときは，まず，そのラベルを削除する，という形でやって下さい。でも，アドレスのついていないラベルは，ラベルオーバーにならない限り別に消す必要もないと思います。

省略形

今あげた11種類のコマンドは，ほとんど省略形を使えるようになっています。最高の省略形については，**第2—4表**を参照して下さい。

プログラム

プログラムについては，サブルーチンだけプログラムの横に簡単な説明をつけるにとどめました。

注意　このプログラムではモニタのほかにLEVEL-II　BASICインタプリタのサブルーチンも使用していますから，LEVEL-II　BASIC　ROMを実装する必要があります。また，LEVEL-I　BASICインタプリタは，逆アセンブルできないことになります。

このプログラムの不便なところ，またはこうしたいと思うところがあれば，まだ9E3Bから先が少し使えますから，拡張も考えています。また，拡張しやすいようにBFCMは最後にもってきましたから，自由にコマンドをつけたすことができます（ただし使える範囲はスタックの関係から9F36までです）。

ラベルは今のところ208個分とってありますが，256個までは拡張できます。しかしラベルを拡張するとプログラムに使える領域が少なくなってしまうので，やめたほうがいいでしょう（TK-M20Kでももっている人なら別ですが）。

ORG	O.
DSA	D.
CON	C.
LAB	L.
PRG	P.
CLR	CL.
HDS	H.
DEL	DE.
NAL	N.
NEW	NE.
MOV	M.

《第2—4表》

言いわすれましたが，ラベルは5文字以内です。

もしかしたら説明不足でわからないことがあるかもしれません，実際にどんどん活用してみてください。

TK-80BS用アセンブラプログラム・リスト

メインプログラム

```
9000 3100A0 START LXI  SP,STACK
9003 3E56         MVI  A,56H
9005 322C86       STA  862CH
9008 3EC3         MVI  A,C3H
900A 217B94       LXI  H,NAD
900D 32DA83       STA  83DAH
9010 22DB83       SHLD 83DBH
9013 21AA91       LXI  H,LAB1
9016 22E28B       SHLD ADR1
9019 AF           XRA  A
901A 32FA8B       STA  COUNT
901D CD6CFA       CALL FA6CH
9020 216F9D       LXI  H,BFTTL
9023 CD04DF       CALL DF04H
9026 3E5C   A1    MVI  A,5CH
9028 CD23D0       CALL D023H
902B CD829B       CALL KEY
902E CDEC9B       CALL CLCH
9031 CA2690       JZ   A1
9034 21039E       LXI  H,BFCOM
9037 0600         MVI  B,00H
9039 111D84 A2    LXI  D,841DH
903C CDCEED       CALL EDCEH
903F 0E03         MVI  C,03H
9041 78           MOV  A,B
9042 BE           CMP  M
9043 CA5890       JZ   A3
9046 1A     A4    LDAX D
9047 BE           CMP  M
9048 13           INX  D
9049 23           INX  H
904A C26E90       JNZ  A5
904D 0D           DCR  C
904E C24690       JNZ  A4
9051 4E     A6    MOV  C,M
9052 23           INX  H
9053 46           MOV  B,M
9054 C5           PUSH B
9055 C3EC9B       JMP  CLCH
9058 CD5C9B A3    CALL TKAD
905B C26D94       JNZ  A8
905E 1A           LDAX D
905F FE3A         CPI  3AH
9061 C26D94       JNZ  A8
9064 13           INX  D
9065 22E68B       SHLD ADR3
9068 CDAC94       CALL LASS
906B C32690       JMP  A1
906E 09     A5    DAD  B
906F 23           INX  H
9070 FE2E         CPI  2EH
9072 C23990       JNZ  A2
9075 79           MOV  A,C
9076 FE03         CPI  03H
9078 CA6D94       JZ   A8
907B 2B           DCX  H
907C 2B           DCX  H
907D C35190       JMP  A6
9080 CD589B ORG   CALL CTAD
9083 C27394       JNZ  B2
9086 CDEC9B       CALL CLCH
9089 C27394       JNZ  B2
908C 22E68B       SHLD ADR3
908F 219590       LXI  H,ORG1
9092 22E28B       SHLD ADR1
9095 2AE68B ORG1  LHLD ADR3
9098 CDBE9A       CALL PR1
909B CD21D0       CALL D021H
909E CD829B       CALL KEY
90A1 1A           LDAX D
90A2 FE0D         CPI  0DH
90A4 CA2690       JZ   A1
90A7 CDEC9B       CALL CLCH
90AA CAB390       JZ   B4
90AD CDAC94       CALL LASS
90B0 C39590       JMP  ORG1
90B3 217E84 B4    LXI  H,847EH
90B6 35           DCR  M
90B7 CDF49A       CALL LDSA
90BA C39590       JMP  ORG1
90BD CACF90 DSA   JZ   B5
90C0 CD589B       CALL CTAD
90C3 C27394       JNZ  B2
90C6 CDEC9B       CALL CLCH
90C9 C27394       JNZ  B2
90CC C3D890       JMP  B6
90CF 2AE08B B5    LHLD STRAD
90D2 7C           MOV  A,H
90D3 A5           ANA  L
90D4 3C           INR  A
90D5 CA6D94       JZ   A8
90D8 22E68B B6    SHLD ADR3
90DB 21E190       LXI  H,DSA1
90DE 22E28B       SHLD ADR1
90E1 CD6CFA DSA1  CALL FA6CH
90E4 CDF49A B7    CALL LDSA
90E7 3AFC7D       LDA  7DFCH
90EA FE53         CPI  53H
90EC CAE490       JZ   B7
90EF 2A7F84       LHLD 847FH
90F2 7C           MOV  A,H
90F3 85           ADD  L
90F4 FE5F         CPI  5FH
```

メインプログラム

```
90F6 C2E490        JNZ  B7
90F9 3AFE7D B8     LDA  7DFEH
90FC E620          ANI  20H
90FE CAF990        JZ   B8
9101 3AFC7D        LDA  7DFCH
9104 FE53          CPI  53H
9106 CAE490        JZ   B7
9109 FE4C          CPI  4CH
910B CAE490        JZ   B7
910E FE4E          CPI  4EH
9110 CAE190        JZ   DSA1
9113 FE0A          CPI  0AH
9115 CA2690        JZ   A1
9118 FE52          CPI  52H
911A CA3891        JZ   B9
911D FE7F          CPI  7FH
911F CA5B91        JZ   C1
9122 FE20          CPI  20H
9124 CA7691        JZ   C2
9127 CD2DF7        CALL F72DH
912A A7            ANA  A
912B FAF990        JM   B8
912E 07            RLC
912F 07            RLC
9130 07            RLC
9131 07            RLC
9132 67            MOV  H,A
9133 2E00          MVI  L,00H
9135 C3D890        JMP  B6
9138 2AE68B B9     LHLD ADR3
913B 11A0FF        LXI  D,FFA0H
913E EB            XCHG
913F 19            DAD  D
9140 E5     C5     PUSH H
9141 CD8899        CALL CHG
9144 0600          MVI  B,00H
9146 4F            MOV  C,A
9147 09            DAD  B
9148 CD68EE        CALL EE68H
914B FA4091        JM   C5
914E 061E          MVI  B,1EH
9150 E1     T9     POP  H
9151 05            DCR  B
9152 C25091        JNZ  T9
9155 3100A0        LXI  SP,STACK
9158 C3D890        JMP  B6
915B 3E08   C1     MVI  A,08H
915D 327984        STA  8479H
9160 0620          MVI  B,20H
9162 CD26D0 C6     CALL D026H
9165 05            DCR  B
9166 C26291        JNZ  C6
9169 2A7F84        LHLD 847FH
916C 7C            MOV  A,H
916D 85            ADD  L
916E FE7E          CPI  7EH
9170 CA2690        JZ   A1
9173 C3F990        JMP  B8
9176 3E3F   C2     MVI  A,3FH
9178 CD23D0        CALL D023H
917B CD829B        CALL KEY
917E CD5C9B        CALL TKAD
9181 C27691        JNZ  C2
9184 CDEC9B        CALL CLCH
9187 C27691        JNZ  C2
918A C3D890        JMP  B6
918D C26D94 CON    JNZ  A8
9190 2AE28B        LHLD ADR1
9193 E5            PUSH H
9194 2AE48B        LHLD ADR2
9197 C9            RET
9198 C26D94 LAB    JNZ  A8
919B 21AA91        LXI  H,LAB1
919E 22E28B        SHLD ADR1
91A1 3ADD8B        LDA  LBCNT
91A4 32FA8B        STA  COUNT
91A7 212D86        LXI  H,LABEL
91AA 3AFA8B LAB1   LDA  COUNT
91AD A7            ANA  A
91AE CA2690        JZ   A1
91B1 CD83D0        CALL D083H
91B4 22E48B        SHLD ADR2
91B7 DA2690        JC   A1
91BA 3AFC7D C7     LDA  7DFCH
91BD FE20          CPI  20H
91BF CABA91        JZ   C7
91C2 CD829A        CALL LBL1
91C5 F5            PUSH PSW
91C6 CD21D0        CALL D021H
91C9 5E            MOV  E,M
91CA 23            INX  H
91CB 56            MOV  D,M
91CC 23            INX  H
91CD E5            PUSH H
91CE 2B            DCX  H
91CF 3E05          MVI  A,05H
91D1 CD07DF        CALL DF07H
91D4 21CB9D        LXI  H,BFEQU
91D7 CD04DF        CALL DF04H
91DA E1            POP  H
91DB F1            POP  PSW
91DC E5            PUSH H
91DD CAF391        JZ   C8
91E0 EB            XCHG
91E1 CDBE9A        CALL PR1
91E4 EB            XCHG
91E5 CD9C97        CALL J9
91E8 CD92ED C9     CALL ED92H
91EB E1            POP  H
91EC 010500        LXI  B,0005H
91EF 09            DAD  B
91F0 C3AA91        JMP  LAB1
91F3 21F69D C8     LXI  H,BF-
91F6 CD04DF        CALL DF04H
91F9 C3E891        JMP  C9
91FC CA2392 PRG    JZ   D1
91FF CD589B        CALL CTAD
9202 C26D94        JNZ  A8
9205 E5            PUSH H
9206 CD589B        CALL CTAD
9209 C2A694        JNZ  B1
920C CDEC9B        CALL CLCH
920F C2A694        JNZ  B1
9212 D1            POP  D
9213 CD68EE        CALL EE68H
9216 DA7094        JC   B3
9219 22DE8B        SHLD ENDAD
921C EB            XCHG
921D 22E08B        SHLD STRAD
9220 C32690        JMP  A1
9223 2AE08B D1     LHLD STRAD
9226 7C            MOV  A,H
9227 A5            ANA  L
9228 3C            INR  A
9229 CA4392        JZ   A7
922C CD21D0        CALL D021H
922F CDBE9A        CALL PR1
9232 3E2D          MVI  A,2DH
9234 CD23D0        CALL D023H
9237 2ADE8B        LHLD ENDAD
923A CDBE9A        CALL PR1
923D CD92ED        CALL ED92H
9240 C32690        JMP  A1
9243 21E99D A7     LXI  H,BFNP
9246 CD04DF        CALL DF04H
9249 C32690        JMP  A1
924C C26D94 CLR    JNZ  A8
924F CD6CFA        CALL FA6CH
9252 C32690        JMP  A1
9255 CD589B HDS    CALL CTAD
9258 C27394        JNZ  B2
925B 22E68B        SHLD ADR3
925E CD589B        CALL CTAD
9261 C27394        JNZ  B2
9264 CDEC9B        CALL CLCH
9267 C27394        JNZ  B2
926A EB            XCHG
926B 2AE68B        LHLD ADR3
926E EB            XCHG
926F CD68EE        CALL EE68H
9272 DA7094        JC   B3
9275 3E55          MVI  A,55H
9277 322C86        STA  862CH
927A EB            XCHG
927B D5     D2     PUSH D
927C CDF49A        CALL LDSA
927F D1            POP  D
9280 CD83D0        CALL D083H
9283 DA9292        JC   9292H
9286 2AE68B        LHLD ADR3
9289 CD7FF7        CALL F77FH
928C DA7B92        JC   D2
928F CA7B92        JZ   D2
9292 3E56          MVI  A,56H
9294 322C86        STA  862CH
9297 C32690        JMP  A1
929A CD589B DEL    CALL CTAD
929D C27394        JNZ  B2
92A0 221684        SHLD 8416H
92A3 CD589B        CALL CTAD
92A6 C27394        JNZ  B2
92A9 221884        SHLD 8418H
92AC CDEC9B        CALL CLCH
92AF C27394        JNZ  B2
92B2 2AE08B        LHLD STRAD
92B5 EB            XCHG
92B6 2A1684        LHLD 8416H
92B9 CD68EE        CALL EE68H
92BC EB            XCHG
92BD DA6D94        JC   A8
92C0 2A1884        LHLD 8418H
92C3 CD68EE        CALL EE68H
92C6 EB            XCHG
92C7 DA7094        JC   B3
92CA CD88F7        CALL F788H
92CD 22F28B        SHLD DEL4
92D0 2ADE8B        LHLD ENDAD
92D3 22F08B        SHLD DEL3
92D6 CD7FF7        CALL F77FH
92D9 DA6D94        JC   A8
92DC CAF992        JZ   D3
```

メインプログラム

```
92DF 13              INX  D
92E0 EB              XCHG
92E1 22EE8B          SHLD DEL2
92E4 EB              XCHG
92E5 2AF08B          LHLD DEL3
92E8 44              MOV  B,H
92E9 4D              MOV  C,L
92EA 2A1684          LHLD 8416H
92ED CD37F9          CALL F937H
92F0 22DE8B          SHLD ENDAD
92F3 CD1F99          CALL MOVE
92F6 C32690          JMP  A1
92F9 2A1684 D3       LHLD 8416H
92FC 2B              DCX  H
92FD EB              XCHG
92FE 2AE08B          LHLD STRAD
9301 EB              XCHG
9302 22DE8B          SHLD ENDAD
9305 CD7FF7          CALL F77FH
9308 D22690          JNC  A1
930B 210000          LXI  H,0000H
930E 22DE8B          SHLD ENDAD
9311 2B              DCX  H
9312 22E08B          SHLD STRAD
9315 C32690          JMP  A1
9318 C26D94 NAL      JNZ  A8
931B 212A93          LXI  H,NAL1
931E 22E28B          SHLD ADR1
9321 3ADD8B          LDA  LBCNT
9324 32FA8B          STA  COUNT
9327 212D86          LXI  H,LABEL
932A 3AFA8B NAL1     LDA  COUNT
932D A7              ANA  A
932E CA2690          JZ   A1
9331 CD829A          CALL LBL1
9334 CA3E93          JZ   D4
9337 010700 D5       LXI  B,0007H
933A 09              DAD  B
933B C32A93          JMP  NAL1
933E CD21D0 D4       CALL D021H
9341 CDAC9A          CALL LBLCL
9344 CD0298          CALL LBLPR
9347 3E3F            MVI  A,3FH
9349 CD23D0          CALL D023H
934C CD829B          CALL KEY
934F 1A              LDAX D
9350 FE0D            CPI  0DH
9352 22E48B          SHLD ADR2
9355 CA2690          JZ   A1
9358 CDEC9B          CALL CLCH
935B CA3793          JZ   D5
935E E5              PUSH H
935F CD5C9B          CALL TKAD
9362 C27C93          JNZ  D6
9365 CDEC9B          CALL CLCH
9368 C27C93          JNZ  D6
936B 7C              MOV  A,H
936C B5              ORA  L
936D CA7C93          JZ   D6
9370 EB              XCHG
9371 E1              POP  H
9372 73              MOV  M,E
9373 23              INX  H
9374 72              MOV  M,D
9375 2B              DCX  H
9376 CD7B98          CALL LBLI1
9379 C33793          JMP  D5
937C 3E07   D6       MVI  A,07H
937E CD04F8          CALL F804H
9381 E1              POP  H
9382 C33E93          JMP  D4
9385 C26D94 NEW      JNZ  A8
9388 06FF            MVI  B,FFH
938A 11E18B          LXI  D,8BE1H
938D 78              MOV  A,B
938E 12              STAX D
938F 1B              DCX  D
9390 12              STAX D
9391 212D86          LXI  H,LABEL
9394 04              INR  B
9395 70     D7       MOV  M,B
9396 23              INX  H
9397 7D              MOV  A,L
9398 BB              CMP  E
9399 C29593          JNZ  D7
939C 7C              MOV  A,H
939D BA              CMP  D
939E C29593          JNZ  D7
93A1 C32690          JMP  A1
93A4 CD589B MOV      CALL CTAD
93A7 C27394          JNZ  B2
93AA 22EE8B          SHLD DEL2
93AD CD589B          CALL CTAD
93B0 C27394          JNZ  B2
93B3 22F08B          SHLD DEL3
93B6 CD589B          CALL CTAD
93B9 C27394          JNZ  B2
93BC 221684          SHLD 8416H
93BF CDEC9B          CALL CLCH
93C2 C27394          JNZ  B2
93C5 2AE08B          LHLD STRAD
93C8 EB              XCHG
93C9 2AEE8B          LHLD DEL2
93CC CD68EE          CALL EE68H
93CF EB              XCHG
93D0 DA6D94          JC   A8
93D3 06FF            MVI  B,FFH
93D5 C2DA93          JNZ  D8
93D8 067F            MVI  B,7FH
93DA 2AF08B D8       LHLD DEL3
93DD CD68EE          CALL EE68H
93E0 EB              XCHG
93E1 DA7094          JC   B3
93E4 2ADE8B          LHLD ENDAD
93E7 CD7FF7          CALL F77FH
93EA DA6D94          JC   A8
93ED 3EFE            MVI  A,FEH
93EF CAF393          JZ   D9
93F2 3C              INR  A
93F3 A0     D9       ANA  B
93F4 47              MOV  B,A
93F5 2AEE8B          LHLD DEL2
93F8 EB              XCHG
93F9 2A1684          LHLD 8416H
93FC E5              PUSH H
93FD CD7FF7          CALL F77FH
9400 22F28B          SHLD DEL4
9403 D1              POP  D
9404 D23F94          JNC  E1
9407 CD88F7          CALL F788H
940A 22F28B          SHLD DEL4
940D 2AE08B          LHLD STRAD
9410 EB              XCHG
9411 CD68EE          CALL EE68H
9414 D21A94          JNC  E2
9417 22E08B          SHLD STRAD
941A E5     E2       PUSH H
941B 78              MOV  A,B
941C E601            ANI  01H
941E 2AF08B          LHLD DEL3
9421 EB              XCHG
9422 2AF28B          LHLD DEL4
9425 19              DAD  D
9426 C22C94          JNZ  T6
9429 22DE8B          SHLD ENDAD
942C 2AEE8B T6       LHLD DEL2
942F EB              XCHG
9430 2AF08B          LHLD DEL3
9433 44              MOV  B,H
9434 4D              MOV  C,L
9435 E1              POP  H
9436 CD37F9          CALL F937H
9439 CD1F99 E3       CALL MOVE
943C C32690          JMP  A1
943F 78     E1       MOV  A,B
9440 A7              ANA  A
9441 EB              XCHG
9442 44              MOV  B,H
9443 4D              MOV  C,L
9444 FA4A94          JM   E4
9447 22E08B          SHLD STRAD
944A 2AF28B E4       LHLD DEL4
944D EB              XCHG
944E 2AF08B          LHLD DEL3
9451 19              DAD  D
9452 EB              XCHG
9453 2ADE8B          LHLD ENDAD
9456 CD7FF7          CALL F77FH
9459 EB              XCHG
945A D26094          JNC  E5
945D 22DE8B          SHLD ENDAD
9460 2AEE8B E5       LHLD DEL2
9463 EB              XCHG
9464 2AF08B          LHLD DEL3
9467 CD22F9          CALL F922H
946A C33994          JMP  E3
946D 3E07   A8       MVI  A,07H
946F 113E02          LXI  D,023EH
9472 113E01          LXI  D,013EH
9475 CD04F8          CALL F804H
9478 C32690          JMP  A1
947B 21969D NAD      LXI  H,BFST
947E CD04DF          CALL DF04H
9481 E1              POP  H
9482 E5              PUSH H
9483 2B              DCX  H
9484 CDBE9A          CALL PR1
9487 219F9D          LXI  H,BFNA
948A CD04DF          CALL DF04H
948D E1              POP  H
948E 23              INX  H
948F 7E              MOV  A,M
9490 CD0298          CALL LBLPR
9493 2A7F84          LHLD 847FH
9496 CD21D0          CALL D021H
9499 2B     E6       DCX  H
949A 7E              MOV  A,M
949B FE20            CPI  20H
949D CA9994          JZ   E6
94A0 23              INX  H
94A1 3622            MVI  M,22H
94A3 CD92ED          CALL ED92H
94A6 3100A0 B1       LXI  SP,STACK
94A9 C32690          JMP  A1
```

サブルーチンプログラム

```
94AC 2AE68B LASS   LHLD ADR3    1行アセンブルするルーチン
94AF 22EC8B        SHLD ADR6
94B2 23            INX  H
94B3 7E            MOV  A,M
94B4 2B            DCX  H
94B5 6E            MOV  L,M
94B6 67            MOV  H,A
94B7 22F68B        SHLD DAT3
94BA 62            MOV  H,D
94BB 6B            MOV  L,E
94BC 22EA8B        SHLD ADR5
94BF CDEC9B        CALL CLCH
94C2 FE22          CPI  22H
94C4 CA6D95        JZ   E7
94C7 FE27          CPI  27H
94C9 CA6D95        JZ   E7
94CC FE2B          CPI  2BH
94CE CA8F95        JZ   T5
94D1 D5            PUSH D
94D2 21009C        LXI  H,BFISR
94D5 AF            XRA  A
94D6 47            MOV  B,A
94D7 32F48B E8     STA  DAT1
94DA D1            POP  D
94DB D5            PUSH D
94DC 7E            MOV  A,M
94DD 32F58B        STA  DAT2
94E0 23            INX  H
94E1 E607          ANI  07H
94E3 CAA695        JZ   E9
94E6 4F            MOV  C,A
94E7 1A     F1     LDAX D
94E8 BE            CMP  M
94E9 C2A595        JNZ  F2
94EC 13            INX  D
94ED 23            INX  H
94EE 0D            DCR  C
94EF C2E794        JNZ  F1
94F2 3AF48B        LDA  DAT1
94F5 FEF4          CPI  F4H
94F7 C20795        JNZ  T3
94FA 1A            LDAX D
94FB FE49          CPI  49H
94FD C20795        JNZ  T3
9500 13            INX  D
9501 21FE13        LXI  H,13FEH
9504 22F48B        SHLD DAT1
9507 E1     T3     POP  H
9508 21F58B        LXI  H,DAT2
950B 7E            MOV  A,M
950C 2B            DCX  H
950D 07            RLC
950E 07            RLC
950F 07            RLC
9510 E607          ANI  07H
9512 CAEE95        JZ   F3
9515 3D            DCR  A
9516 C22295        JNZ  T1
9519 CDCB9B        CALL CHEC3
951C 07     T2     RLC
951D 07            RLC
951E 07            RLC
951F C3EE95        JMP  F3
9522 3D     T1     DCR  A
9523 C23595        JNZ  F5
9526 CDEC9B        CALL CLCH
9529 CD2DF7        CALL F72DH
952C FE08          CPI  08H
952E D2B295        JNC  G4
9531 13            INX  D
9532 C31C95        JMP  T2
9535 3D     F5     DCR  A
9536 CAEB95        JZ   F4
9539 3D            DCR  A
953A C24E95        JNZ  F7
```

```
953D CDCB9B        CALL CHEC3
9540 07            RLC
9541 07            RLC
9542 07            RLC
9543 86            ADD  M
9544 77            MOV  M,A
9545 CDF29B        CALL CHECC
9548 CAEB95        JZ   F4
954B C3B295        JMP  G4
954E 3D     F7     DCR  A
954F C25895        JNZ  F8
9552 CD889B        CALL CHEC1
9555 C3EE95        JMP  F3
9558 3D     F8     DCR  A
9559 C26295        JNZ  F9
955C CD9C9B        CALL CHEC2
955F C3EE95        JMP  F3
9562 CD889B F9     CALL CHEC1
9565 FE20          CPI  20H
9567 DAEE95        JC   F3
956A C3B295        JMP  G4
956D 47     E7     MOV  B,A
956E 13     G1     INX  D
956F CDA6DE        CALL DEA6H
9572 CACC95        JZ   F6
9575 B8            CMP  B
9576 CA8795        JZ   G2
9579 4F            MOV  C,A
957A 78            MOV  A,B
957B FE27          CPI  27H
957D CC779B        CZ   AJCH
9580 79            MOV  A,C
9581 CD0D97        CALL MEMIN
9584 C36E95        JMP  G1
9587 13     G2     INX  D
9588 CDEC9B        CALL CLCH
958B C8            RZ
958C C3CC95        JMP  F6
958F 13     T5     INX  D
9590 CD6F9B        CALL TKDA
9593 C2CC95        JNZ  F6
9596 CD0D97        CALL MEMIN
9599 CDEC9B        CALL CLCH
959C C8            RZ
959D FE2C          CPI  2CH
959F CA8F95        JZ   T5
95A2 C3CC95        JMP  F6
95A5 09     F2     DAD  B
95A6 3AF48B E9     LDA  DAT1
95A9 CD9D9A        CALL INCR
95AC FE7E          CPI  7EH
95AE C2D794        JNZ  E8
95B1 D1            POP  D
95B2 2AEA8B G4     LHLD ADR5
95B5 EB            XCHG
95B6 2AEC8B        LHLD ADR6
95B9 22E68B        SHLD ADR3
95BC 3AF68B        LDA  DAT3
95BF 77            MOV  M,A
95C0 23            INX  H
95C1 3AF78B        LDA  DAT4
95C4 77            MOV  M,A
95C5 2B            DCX  H
95C6 CD1898        CALL LBLIN
95C9 D2D795        JNC  T4
95CC 2AEC8B F6     LHLD ADR6
95CF 22E68B        SHLD ADR3
95D2 3E07          MVI  A,07H
95D4 C304F8        JMP  F804H
95D7 21D19D T4     LXI  H,BFLB
95DA F5            PUSH PSW
95DB CD04DF        CALL DF04H
95DE F1            POP  PSW
95DF 21DA9D        LXI  H,BFIN
95E2 CA04DF        JZ   DF04H
95E5 21E19D        LXI  H,BFDL
95E8 C304DF        JMP  DF04H
```

```
95EB CDCB9B F4     CALL CHEC3
95EE 86       F3   ADD  M
95EF CD0D97        CALL MEMIN
95F2 23            INX  H
95F3 7E            MOV  A,M
95F4 0F            RRC
95F5 0F            RRC
95F6 0F            RRC
95F7 E603          ANI  03H
95F9 3D            DCR  A
95FA CA3B96        JZ   G6
95FD 47            MOV  B,A
95FE 7E            MOV  A,M
95FF E6E0          ANI  E0H
9601 CA0A96        JZ   G7
9604 CDF29B        CALL CHECC
9607 C2B295        JNZ  G4
960A CDCEED G7     CALL EDCEH
960D 05            DCR  B
960E C26296        JNZ  G8
9611 D5            PUSH D
9612 CD6F9B        CALL TKDA
9615 C23F96        JNZ  G9
9618 F5            PUSH PSW
9619 CDEC9B        CALL CLCH
961C FE48          CPI  48H
961E CA3696        JZ   H1
9621 F1            POP  PSW
9622 D1            POP  D
9623 CD24EE        CALL EE24H
9626 DAB295        JC   G4
9629 7D            MOV  A,L
962A D5            PUSH D
962B CD0D97 H2     CALL MEMIN
962E CDEC9B H3     CALL CLCH
9631 D1            POP  D
9632 C8            RZ
9633 C3B295        JMP  G4
9636 13       H1   INX  D
9637 F1            POP  PSW
9638 C32B96        JMP  H2
963B D5       G6   PUSH D
963C C32E96        JMP  H3
963F D1       G9   POP  D
9640 EB            XCHG
9641 7E            MOV  A,M
9642 23            INX  H
9643 4E            MOV  C,M
9644 23            INX  H
9645 BE            CMP  M
9646 C2B295        JNZ  G4
9649 FE27          CPI  27H
964B CA5696        JZ   H4
964E FE22          CPI  22H
9650 CA5996        JZ   H5
9653 C3B295        JMP  G4
9656 CD779B H4     CALL AJCH
9659 79       H5   MOV  A,C
965A CD0D97        CALL MEMIN
965D 23            INX  H
965E EB            XCHG
965F C33B96        JMP  G6
9662 1A       G8   LDAX D
9663 FE24          CPI  24H
9665 CAD496        JZ   H6
9668 D5            PUSH D
9669 CD5C9B        CALL TKAD
966C C27E96        JNZ  H7
966F 1A            LDAX D
9670 FE48          CPI  48H
9672 C27E96        JNZ  H7
9675 13            INX  D
9676 CDEC9B        CALL CLCH
9679 D1            POP  D
967A CA0897        JZ   H9
967D D5            PUSH D
967E D1       H7   POP  D
```

```
967F 1A            LDAX D
9680 FE2F          CPI  2FH
9682 CAB295        JZ   G4
9685 3ADD8B        LDA  LBCNT
9688 32FA8B        STA  COUNT
968B CD359B        CALL CRCH
968E C2B295        JNZ  G4
9691 212686        LXI  H,8626H
9694 010700        LXI  B,0007H
9697 09       I1   DAD  B
9698 3AFA8B        LDA  COUNT
969B A7            ANA  A
969C CAC996        JZ   I2
969F CD829A        CALL LBL1
96A2 23            INX  H
96A3 23            INX  H
96A4 CD099B        CALL COM
96A7 2B            DCX  H
96A8 2B            DCX  H
96A9 C29796        JNZ  I1
96AC 7E            MOV  A,M
96AD 5F            MOV  E,A
96AE 23            INX  H
96AF 56            MOV  D,M
96B0 2B            DCX  H
96B1 A2            ANA  D
96B2 3C            INR  A
96B3 EB            XCHG
96B4 C20897        JNZ  H9
96B7 EB            XCHG
96B8 CDAC9A I4     CALL LBLCL
96BB 47            MOV  B,A
96BC 2AEC8B        LHLD ADR6
96BF 7E            MOV  A,M
96C0 36EF          MVI  M,EFH
96C2 CD0D97        CALL MEMIN
96C5 78            MOV  A,B
96C6 C30D97        JMP  MEMIN
96C9 01FFFF I2     LXI  B,FFFFH
96CC 21B896        LXI  H,I4
96CF E5            PUSH H
96D0 C5            PUSH B
96D1 C3BB98        JMP  I3
96D4 13       H6   INX  D
96D5 CDEC9B        CALL CLCH
96D8 CA0197        JZ   I5
96DB 47            MOV  B,A
96DC 13            INX  D
96DD CDCEED        CALL EDCEH
96E0 CD24EE        CALL EE24H
96E3 DAB295        JC   G4
96E6 CDEC9B        CALL CLCH
96E9 C2B295        JNZ  G4
96EC EB            XCHG
96ED 2AEC8B        LHLD ADR6
96F0 78            MOV  A,B
96F1 FE2B          CPI  2BH
96F3 CA0797        JZ   I6
96F6 FE2D          CPI  2DH
96F8 C2B295        JNZ  G4
96FB CD7FF7        CALL F77FH
96FE C30897        JMP  H9
9701 110000 I5     LXI  D,0000H
9704 2AEC8B        LHLD ADR6
9707 19       I6   DAD  D
9708 7D       H9   MOV  A,L
9709 CD0D97        CALL MEMIN
970C 7C            MOV  A,H
970D D5      MEMIN PUSH D     Accの内容をADR3の
970E E5            PUSH H     ドレスに入れ，ADR3を
970F 2AE68B        LHLD ADR3  リメントする
9712 77            MOV  M,A
9713 EB            XCHG
9714 2AE08B        LHLD STRAD
9717 EB            XCHG
9718 CD68EE        CALL EE68H
971B D22197        JNC  I8
```

```
971E 22E08B         SHLD STRAD
9721 EB      I8     XCHG
9722 2ADE8B         LHLD ENDAD
9725 CD7FF7         CALL F77FH
9728 EB             XCHG
9729 D22F97         JNC  I9
972C 22DE8B         SHLD ENDAD
972F 23      I9     INX  H
9730 22E68B         SHLD ADR3
9733 E1             POP  H
9734 D1             POP  D
9735 C9             RET
9736 CDFF9A  LDS1   CALL GET      1行逆アセンブル，復帰改行はし
9739 22E88B         SHLD ADR4     ない
973C CDBE9A         CALL PR1
973F CD21D0         CALL D021H
9742 CD8899         CALL CHG
9745 DAE297         JC   J1
9748 57             MOV  D,A
9749 7E             MOV  A,M
974A 4F             MOV  C,A
974B CDC99A         CALL PR2
974E 5A             MOV  E,D
974F 21F88B         LXI  H,BYT2
9752 1D      J2     DCR  E
9753 CA6397         JZ   J3
9756 E5             PUSH H
9757 CDFF9A         CALL GET
975A E1             POP  H
975B 77             MOV  M,A
975C 23             INX  H
975D CDC99A         CALL PR2
9760 C35297         JMP  J2
9763 3E04    J3     MVI  A,04H
9765 92             SUB  D
9766 87             ADD  A
9767 3D             DCR  A
9768 5F             MOV  E,A
9769 CD21D0  J4     CALL D021H
976C 1D             DCR  E
976D C26997         JNZ  J4
9770 CDDA99         CALL LIPR
9773 79             MOV  A,C
9774 15             DCR  D
9775 CAB497         JZ   J5
9778 15             DCR  D
9779 CAA197         JZ   J6
977C 05             DCR  B
977D FA8697         JM   J7
9780 CDE79A         CALL PR3
9783 CDFA9A         CALL CPR
9786 2AF88B  J7     LHLD BYT2
9789 E5             PUSH H
978A CD429A         CALL LBLSR
978D DA9897         JC   J8
9790 3E05           MVI  A,05H
9792 2B             DCX  H
9793 CD07DF         CALL DF07H
9796 E1             POP  H
9797 C9             RET
9798 E1      J8     POP  H
9799 CDBE9A         CALL PR1
979C 3E48    J9     MVI  A,48H
979E C323D0         JMP  D023H
97A1 05      J6     DCR  B
97A2 FAAB97         JM   K1
97A5 CDD99A         CALL PR5
97A8 CDFA9A         CALL CPR
97AB 3AF88B  K1     LDA  BYT2
97AE CDC99A         CALL PR2
97B1 C39C97         JMP  J9
97B4 05      J5     DCR  B
97B5 F8             RM
97B6 CAD99A         JZ   PR5
97B9 05             DCR  B
97BA C2C597         JNZ  K2
97BD 0F             RRC
```

```
97BE 0F             RRC
97BF 0F             RRC
97C0 E637           ANI  37H
97C2 C323D0         JMP  D023H
97C5 05      K2     DCR  B
97C6 CADC9A         JZ   PR4
97C9 05             DCR  B
97CA C2D797         JNZ  K3
97CD CDD99A         CALL PR5
97D0 CDFA9A         CALL CPR
97D3 79             MOV  A,C
97D4 C3DC9A         JMP  PR4
97D7 05      K3     DCR  B
97D8 05             DCR  B
97D9 C2E79A         JNZ  PR3
97DC 21C19D         LXI  H,BFPSW
97DF C3EA9A         JMP  PR6
97E2 CDFF9A  J1     CALL GET
97E5 4F             MOV  C,A
97E6 CDC99A         CALL PR2
97E9 CDFF9A         CALL GET
97EC 5F             MOV  E,A
97ED 21F69D         LXI  H,BF-
97F0 CD04DF         CALL DF04H
97F3 CDDA99         CALL LIPR
97F6 79             MOV  A,C
97F7 05             DCR  B
97F8 FA0198         JM   K4
97FB CDE79A         CALL PR3
97FE CDFA9A         CALL CPR
9801 7B      K4     MOV  A,E
9802 D5      LBLPR  PUSH D        ラベル番号(Acc)から，ラベルを
9803 E5             PUSH H        プリント
9804 110700         LXI  D,0007H
9807 62             MOV  H,D
9808 6F             MOV  L,A
9809 CD95F7         CALL F795H
980C 112E86         LXI  D,862EH
980F 19             DAD  D
9810 3E05           MVI  A,05H
9812 CD07DF         CALL DF07H
9815 E1             POP  H
9816 D1             POP  D
9817 C9             RET
9818 CDCEED  LBLIN  CALL EDCEH    HL＝アドレス，DE＝ラベルの
981B 7C             MOV  A,H      先頭番地で，そのラベルとアドレ
981C B5             ORA  L        スが同じものがあればラベルをと
981D 37             STC           りかえる，アドレスもラベルも同
981E C8             RZ            じならなにもしない，アドレスが
981F 1A             LDAX D        違ってラベルが同じならエラー
9820 FE2F           CPI  2FH      (c＝"1")
9822 CAE198         JZ   K5       ラベルが"1"なら削除
9825 CD359B         CALL CRCH     入力のときはz＝"1"，削除の
9828 D8             RC            ときはz＝"0"
9829 C5             PUSH B
982A 44             MOV  B,H
982B 4D             MOV  C,L
982C 78             MOV  A,B
982D A1             ANA  C
982E 3C             INR  A
982F CABB98         JZ   I3
9832 3ADD8B         LDA  LBCNT
9835 32FA8B         STA  COUNT
9838 212D86         LXI  H,LABEL
983B 3AFA8B  K6     LDA  COUNT
983E A7             ANA  A
983F CABB98         JZ   I3
9842 CD829A         CALL LBL1
9845 CA6698         JZ   K7
9848 79             MOV  A,C
9849 BE             CMP  M
984A 23             INX  H
984B C25C98         JNZ  K8
984E 78             MOV  A,B
984F BE             CMP  M
9850 C25C98         JNZ  K8
9853 23             INX  H
```

```
9854 CD099B          CALL COM
9857 C2D598          JNZ  K9
985A C1              POP  B
985B C9              RET
985C 23       K8     INX  H
985D CD099B          CALL COM
9860 C26E98          JNZ  L1
9863 C1       L2     POP  B
9864 37              STC
9865 C9              RET
9866 23       K7     INX  H
9867 23              INX  H
9868 CD099B          CALL COM
986B CA7698          JZ   L3
986E 23       L1     INX  H
986F 23              INX  H
9870 23              INX  H
9871 23              INX  H
9872 23              INX  H
9873 C33B98          JMP  K6
9876 2B       L3     DCX  H
9877 70              MOV  M,B
9878 2B              DCX  H
9879 71              MOV  M,C
987A C1              POP  B
987B C5       LBLI1  PUSH B
987C D5              PUSH D
987D E5              PUSH H
987E CDAC9A          CALL LBLCL
9881 4F              MOV  C,A
9882 2AE08B          LHLD STRAD
9885 CD8899   L4     CALL CHG
9888 DA9F98          JC   L5
988B 5F              MOV  E,A
988C 1600            MVI  D,00H
988E 19              DAD  D
988F EB       L6     XCHG
9890 2ADE8B   L7     LHLD ENDAD
9893 CD68EE          CALL EE68H
9896 EB              XCHG
9897 D28598          JNC  L4
989A AF              XRA  A
989B E1              POP  H
989C D1              POP  D
989D C1              POP  B
989E C9              RET
989F 23       L5     INX  H
98A0 23              INX  H
98A1 7E              MOV  A,M
98A2 23              INX  H
98A3 B9              CMP  C
98A4 C28F98          JNZ  L6
98A7 2B              DCX  H
98A8 2B              DCX  H
98A9 7E              MOV  A,M
98AA 2B              DCX  H
98AB 77              MOV  M,A
98AC 23              INX  H
98AD EB              XCHG
98AE E1              POP  H
98AF 7E              MOV  A,M
98B0 12              STAX D
98B1 13              INX  D
98B2 23              INX  H
98B3 7E              MOV  A,M
98B4 12              STAX D
98B5 13              INX  D
98B6 2B              DCX  H
98B7 E5              PUSH H
98B8 C39098          JMP  L7
98BB CD1D9B   I3     CALL ZRSR
98BE D2CA98          JNC  L8
98C1 21899D          LXI  H,BFLO
98C4 CD04DF          CALL DF04H
98C7 C3A694          JMP  B1
98CA 71       L8     MOV  M,C
98CB 23              INX  H
```

HLからラベル番号を計算して，プログラム中の3バイト命令でそのラベル番号があればそこにアドレスを入れる。

```
98CC 70              MOV  M,B
98CD 23              INX  H
98CE 3ADD8B          LDA  LBCNT
98D1 3C              INR  A
98D2 32DD8B          STA  LBCNT
98D5 42       K9     MOV  B,D
98D6 4B              MOV  C,E
98D7 03              INX  B
98D8 03              INX  B
98D9 03              INX  B
98DA 03              INX  B
98DB CD37F9          CALL F937H
98DE C1              POP  B
98DF AF              XRA  A
98E0 C9              RET
98E1 7C       K5     MOV  A,H
98E2 A5              ANA  L
98E3 3C              INR  A
98E4 37              STC
98E5 C8              RZ
98E6 13              INX  D
98E7 CDA6DE          CALL DEA6H
98EA 37              STC
98EB C0              RNZ
98EC 3ADD8B          LDA  LBCNT
98EF 32FA8B          STA  COUNT
98F2 44              MOV  B,H
98F3 4D              MOV  C,L
98F4 212D86          LXI  H,LABEL
98F7 110700          LXI  D,0007H
98FA 3AFA8B   L9     LDA  COUNT
98FD A7              ANA  A
98FE 37              STC
98FF C8              RZ
9900 CD829A          CALL LBL1
9903 CA1299          JZ   M1
9906 79              MOV  A,C
9907 BE              CMP  M
9908 C21299          JNZ  M1
990B 23              INX  H
990C 78              MOV  A,B
990D BE              CMP  M
990E 2B              DCX  H
990F CA1699          JZ   M2
9912 19       M1     DAD  D
9913 C3FA98          JMP  L9
9916 72       M2     MOV  M,D
9917 23              INX  H
9918 72              MOV  M,D
9919 21DD8B          LXI  H,LBCNT
991C 35              DCR  M
991D 14              INR  D
991E C9              RET
991F 2AE08B   MOVE   LHLD STRAD
9922 CD8899   M3     CALL CHG
9925 DA3799          JC   M4
9928 FE03            CPI  03H
992A C23799          JNZ  M4
992D E5              PUSH H
992E 44              MOV  B,H
992F 4D              MOV  C,L
9930 03              INX  B
9931 CD6899          CALL MOV1
9934 E1              POP  H
9935 3E03            MVI  A,03H
9937 0600     M4     MVI  B,00H
9939 4F              MOV  C,A
993A 09              DAD  B
993B EB              XCHG
993C 2ADE8B          LHLD ENDAD
993F EB              XCHG
9940 CD68EE          CALL EE68H
9943 DA2299          JC   M3
9946 3ADD8B          LDA  LBCNT
9949 32FA8B          STA  COUNT
994C 212D86          LXI  H,LABEL
994F 3AFA8B   M5     LDA  COUNT
```

プログラムを移動させるのではなく，移動させたあとにアドレスを合わせる

```
9952 A7            ANA  A
9953 C8            RZ
9954 CD829A        CALL LBL1
9957 CA6199        JZ   M6
995A 44            MOV  B,H
995B 4D            MOV  C,L
995C CD6899        CALL MOV1
995F 60            MOV  H,B
9960 69            MOV  L,C
9961 010700 M6     LXI  B,0007H
9964 09            DAD  B
9965 C34F99        JMP  M5
9968 0A     MOV1   LDAX B          DE←〔BC〕〔BC+1〕，DEが
9969 5F            MOV  E,A        DEL2とDEL3の間にあれば，
996A 03            INX  B          DE←DE+DEL4，
996B 0A            LDAX B          〔BC〕〔BC+1〕←DE
996C 57            MOV  D,A
996D 0B            DCX  B
996E 2AEE8B        LHLD DEL2
9971 2B            DCX  H
9972 CD7FF7        CALL F77FH
9975 D0            RNC
9976 2AF08B        LHLD DEL3
9979 CD7FF7        CALL F77FH
997C D8            RC
997D 2AF28B        LHLD DEL4
9980 19            DAD  D
9981 7D            MOV  A,L
9982 02            STAX B
9983 03            INX  B
9984 7C            MOV  A,H
9985 02            STAX B
9986 0B            DCX  B
9987 C9            RET
9988 7E     CHG    MOV  A,M        〔HL〕の1バイトを解読し，
9989 FEEF          CPI  EFH        B←TYPE，A←BYTEを入
998B C29999        JNZ  CHG1       れる。ただし〔HL〕=EFの場合
998E 23            INX  H          は，〔HL+1〕が3バイト命令以
998F 7E            MOV  A,M        外ならRST 5として見て，
9990 2B            DCX  H          〔HL+1〕が3バイト命令なら，
9991 CD9999        CALL CHG1       〔HL+1〕の1バイトを解読して，
9994 FE03          CPI  03H        c=“1”でリターン
9996 3F            CMC
9997 D8            RC
9998 7E            MOV  A,M
9999 E5     CHG1   PUSH H          〔HL〕の1バイトを解読し，B←
999A D5            PUSH D          TYPE，A←BYTEを入れる
999B C5            PUSH B
999C 21009C        LXI  H,BFISR
999F FE40          CPI  40H
99A1 FAB099        JM   M7
99A4 FE76          CPI  76H
99A6 CAB099        JZ   M7
99A9 E6F8          ANI  F8H
99AB FAB099        JM   M7
99AE 3E40          MVI  A,40H
99B0 5F     M7     MOV  E,A
99B1 0600          MVI  B,00H
99B3 50            MOV  D,B
99B4 7B     M8     MOV  A,E
99B5 BA            CMP  D
99B6 7E            MOV  A,M
99B7 CAC799        JZ   M9
99BA E607          ANI  07H
99BC 4F            MOV  C,A
99BD 09            DAD  B
99BE 23            INX  H
99BF 7A            MOV  A,D
99C0 CD9D9A        CALL INCR
99C3 57            MOV  D,A
99C4 C3B499        JMP  M8
99C7 07     M9     RLC
99C8 07            RLC
99C9 07            RLC
99CA E607          ANI  07H
99CC 57            MOV  D,A
99CD 7E            MOV  A,M
99CE 0F            RRC
99CF 0F            RRC
99D0 0F            RRC
99D1 E603          ANI  03H
99D3 C1            POP  B
99D4 42            MOV  B,D
99D5 D1            POP  D
99D6 E1            POP  H
99D7 C0            RNZ
99D8 3C            INR  A
99D9 C9            RET
99DA 2AE88B LIPR   LHLD ADR4       ADR4からラベルをプリント，
99DD CD429A        CALL LBLSR      1マスあけて〔ADR4〕のインス
99E0 3E05          MVI  A,05H      トラクションをプリント
99E2 2B            DCX  H
99E3 CD07DF        CALL DF07H
99E6 CD21D0        CALL D021H
99E9 C5            PUSH B
99EA D5            PUSH D
99EB 78            MOV  A,B
99EC A7            ANA  A
99ED CA0E9A        JZ   N1
99F0 D603          SUI  03H
99F2 D2FB99        JNC  N2
99F5 79            MOV  A,C
99F6 E6C7          ANI  C7H
99F8 C30D9A        JMP  N3
99FB C2049A N2     JNZ  N4
99FE 79            MOV  A,C
99FF E6F8          ANI  F8H
9A01 C30D9A        JMP  N3
9A04 3D     N4     DCR  A
9A05 3E40          MVI  A,40H
9A07 CA0D9A        JZ   N3
9A0A 79            MOV  A,C
9A0B E6CF          ANI  CFH
9A0D 4F     N3     MOV  C,A
9A0E 21009C N1     LXI  H,BFISR
9A11 1600          MVI  D,00H
9A13 42            MOV  B,D
9A14 79     N5     MOV  A,C
9A15 B8            CMP  B
9A16 7E            MOV  A,M
9A17 CA279A        JZ   N6
9A1A 23            INX  H
9A1B E607          ANI  07H
9A1D 5F            MOV  E,A
9A1E 19            DAD  D
9A1F 78            MOV  A,B
9A20 CD9D9A        CALL INCR
9A23 47            MOV  B,A
9A24 C3149A        JMP  N5
9A27 E607   N6     ANI  07H
9A29 C2319A        JNZ  N7
9A2C 3E03          MVI  A,03H
9A2E 21FF9D        LXI  H,9DFFH
9A31 CD07DF N7     CALL DF07H
9A34 D606          SUI  06H
9A36 2F            CMA
9A37 47            MOV  B,A
9A38 CD21D0 N8     CALL D021H
9A3B 05            DCR  B
9A3C C2389A        JNZ  N8
9A3F D1            POP  D
9A40 C1            POP  B
9A41 C9            RET
9A42 C5     LBLSR  PUSH B          HLのアドレスをもつラベルをさ
9A43 D5            PUSH D          がし，あれば〔HL〕←ラベルの先
9A44 7C            MOV  A,H        頭番地，なければ〔HL〕←BFB
9A45 B5            ORA  L          L5の先頭番地を入れてリターン
9A46 CA7B9A        JZ   N9
9A49 7C            MOV  A,H
9A4A A5            ANA  L
9A4B 3C            INR  A
9A4C CA7B9A        JZ   N9
9A4F EB            XCHG
9A50 212686        LXI  H,8626H
```

```
9A53 3ADD8B          LDA   LBCNT
9A56 32FA8B          STA   COUNT
9A59 010700          LXI   B,0007H
9A5C 09        P1    DAD   B
9A5D 3AFA8B          LDA   COUNT
9A60 A7              ANA   A
9A61 CA7B9A          JZ    N9
9A64 CD829A          CALL  LBL1
9A67 CA5C9A          JZ    P1
9A6A 7B              MOV   A,E
9A6B BE              CMP   M
9A6C C25C9A          JNZ   P1
9A6F 23              INX   H
9A70 7A              MOV   A,D
9A71 BE              CMP   M
9A72 2B              DCX   H
9A73 C25C9A          JNZ   P1
9A76 23              INX   H
9A77 23              INX   H
9A78 D1              POP   D
9A79 C1              POP   B
9A7A C9              RET
9A7B 21FB9D    N9    LXI   H,BFBL5
9A7E 37              STC
9A7F D1              POP   D
9A80 C1              POP   B
9A81 C9              RET
9A82 D5        LBL1  PUSH  D        HLがラベルバッファの中にあり,
9A83 110600          LXI   D,0006H  HLをラベルのあるところまでイ
9A86 7E        P2    MOV   A,M      ンクリメント, そのラベルにアド
9A87 23              INX   H        レスがきまっていなければz="1"
9A88 B6              ORA   M        にしてリターン
9A89 19              DAD   D
9A8A CA869A          JZ    P2
9A8D CD7FF7          CALL  F77FH
9A90 3AFA8B          LDA   COUNT
9A93 3D              DCR   A
9A94 32FA8B          STA   COUNT
9A97 D1              POP   D
9A98 7E              MOV   A,M
9A99 2B              DCX   H
9A9A A6              ANA   M
9A9B 3C              INR   A
9A9C C9              RET
9A9D FE40      INCR  CPI   40H      Accの値を00から順に, 00,
9A9F FAA79A          JM    P3       01, 02, ……, 3F, 40,
9AA2 07              RLC            80, 88, 90, ……, B8,
9AA3 C8              RZ             C0, C1, C2, ……, FF,
9AA4 0F              RRC            76とふやす
9AA5 C607            ADI   07H
9AA7 3C        P3    INR   A
9AA8 C0              RNZ
9AA9 3E76            MVI   A,76H
9AAB C9              RET
9AAC D5        LBLCL PUSH  D        HLからラベル番号を計算
9AAD E5              PUSH  H
9AAE 112D86          LXI   D,LABEL
9AB1 CD7FF7          CALL  F77FH
9AB4 110700          LXI   D,0007H
9AB7 CDCCF7          CALL  F7CCH
9ABA 7D              MOV   A,L
9ABB E1              POP   H
9ABC D1              POP   D
9ABD C9              RET
9ABE F5        PR1   PUSH  PSW      HLの値をプリント
9ABF 7C              MOV   A,H
9AC0 CDC99A          CALL  PR2
9AC3 7D              MOV   A,L
9AC4 CDC99A          CALL  PR2
9AC7 F1              POP   PSW
9AC8 C9              RET
9AC9 E5        PR2   PUSH  H        Accの値をプリント
9ACA F5              PUSH  PSW
9ACB CD4CF7          CALL  F74CH
9ACE 3E02            MVI   A,02H
9AD0 217284          LXI   H,8472H
9AD3 CD07DF          CALL  DF07H
```

```
9AD6 F1              POP   PSW
9AD7 E1              POP   H
9AD8 C9              RET
9AD9 0F        PR5   RRC            Accのbit3, 4, 5, からB,
9ADA 0F              RRC            C, D, E, H, L, M, Aのど
9ADB 0F              RRC            れかをプリント
9ADC E607      PR4   ANI   07H      Accのbit0, 1, 2からB,
9ADE 21B09D          LXI   H,BFBC   C, D, E, H, L, M, Aのど
9AE1 85              ADD   L        れかをプリント
9AE2 6F              MOV   L,A
9AE3 7E              MOV   A,M
9AE4 C323D0          JMP   D023H
9AE7 21B89D    PR3   LXI   H,BFSP   Accのbit4, 5からB, D,
9AEA 0F        PR6   RRC            H, SPのどれかをプリント
9AEB 0F              RRC
9AEC 0F              RRC
9AED E606            ANI   06H
9AEF 85              ADD   L
9AF0 6F              MOV   L,A
9AF1 C304DF          JMP   DF04H
9AF4 CD3697    LDSA  CALL  LDS1     1行逆アセンブルするルーチン
9AF7 C392ED          JMP   ED92H
9AFA 3E2C      CPR   MVI   A,2CH    「,」を一つプリント
9AFC C323D0          JMP   D023H
9AFF 2AE68B    GET   LHLD  ADR3     ADR3の指す1バイトをとって
9B02 7E              MOV   A,M      きて(ADR3)をインクリメント
9B03 23              INX   H
9B04 22E68B          SHLD  ADR3
9B07 2B              DCX   H
9B08 C9              RET
9B09 C5        COM   PUSH  B        (DE)から5バイトと(HL)から
9B0A D5              PUSH  D        5バイトをくらべる
9B0B E5              PUSH  H        すべて同じ場合のみ, z= "1"
9B0C 0605            MVI   B,05H
9B0E 1A        P4    LDAX  D
9B0F BE              CMP   M
9B10 13              INX   D
9B11 23              INX   H
9B12 C2199B          JNZ   P5
9B15 05              DCR   B
9B16 C20E9B          JNZ   P4
9B19 E1        P5    POP   H
9B1A D1              POP   D
9B1B C1              POP   B
9B1C C9              RET
9B1D D5        ZRSR  PUSH  D        ラベルを入れるためにあいている
9B1E 110700          LXI   D,0007H  ところをさがす
9B21 212D86          LXI   H,LABEL
9B24 7E        P6    MOV   A,M
9B25 23              INX   H
9B26 B6              ORA   M
9B27 2B              DCX   H
9B28 CA339B          JZ    P7
9B2B 19              DAD   D
9B2C 7D              MOV   A,L
9B2D FEDD            CPI   DDH
9B2F C2249B          JNZ   P6
9B32 37              STC
9B33 D1        P7    POP   D
9B34 C9              RET
9B35 1A        CRCH  LDAX  D        ラベルが5文字より短い場合, た
9B36 FE0E            CPI   0EH      とえば "COM" のとき ASCII
9B38 D8              RC             で43, 4F, 4D, 0D, 0A
9B39 C5              PUSH  B        となっているのを43, 4F,
9B3A D5              PUSH  D        4D, 20, 20となおす
9B3B 13              INX   D
9B3C 0605            MVI   B,05H
9B3E 1A        P8    LDAX  D
9B3F FE0D            CPI   0DH
9B41 3E20            MVI   A,20H
9B43 CA519B          JZ    P9
9B46 13              INX   D
9B47 05              DCR   B
9B48 C23E9B          JNZ   P8
9B4B 37              STC
9B4C D1        Q1    POP   D
9B4D C1              POP   B
```

```
9B4E C9               RET
9B4F 12        Q2     STAX D
9B50 13               INX  D
9B51 05        P9     DCR  B
9B52 C24F9B           JNZ  Q2
9B55 C34C9B           JMP  Q1
9B58 CDF29B   CTAD    CALL CHECC     「,」を一つとアドレスを一つとる
9B5B D8               RC
9B5C CDCEED   TKAD    CALL EDCEH     〔DE〕から16進4桁までのアド
9B5F D5               PUSH D         レスをとってHLに入れる
9B60 EB               XCHG
9B61 111484           LXI  D,8414H
9B64 CDA6F6           CALL F6A6H
9B67 A7               ANA  A
9B68 D1               POP  D
9B69 C0               RNZ
9B6A EB               XCHG
9B6B 2A1484           LHLD 8414H
9B6E C9               RET
9B6F CD5C9B   TKDA    CALL TKAD      〔DE〕から16進2桁までのデー
9B72 C0               RNZ            タをとってAccに入れる
9B73 AF               XRA  A
9B74 B4               ORA  H
9B75 7D               MOV  A,L
9B76 C9               RET
9B77 79       AJCH    MOV  A,C       Cレジスタに入っているASCI
9B78 D640             SUI  40H       Iコードを JISコードに変換す
9B7A 4F               MOV  C,A       る
9B7B F0               RP
9B7C C640             ADI  40H
9B7E E67F             ANI  7FH
9B80 4F               MOV  C,A
9B81 C9               RET
9B82 111D84   KEY     LXI  D,841DH   キー入力して，DE←841Dに
9B85 C346F9           JMP  F946H     セット（実際は順序が逆）
9B88 CDCEED   CHEC1   CALL EDCEH     〔DE〕から先がB，D，H，SP
9B8B 1A               LDAX D         のどれかか調べる。どれでもなけ
9B8C 13               INX  D         ればキャリーをセットしてリター
9B8D FE53             CPI  53H       ン
9B8F C2A69B           JNZ  Q3
9B92 1A               LDAX D
9B93 13               INX  D
9B94 FE50             CPI  50H
9B96 3E30      Q4     MVI  A,30H
9B98 C8               RZ
9B99 C3E29B           JMP  A9
9B9C CDCEED   CHEC2   CALL EDCEH     〔DE〕から先がB，D，H，PS
9B9F 1A               LDAX D         Wのどれかか調べる。どれでもな
9BA0 13               INX  D         ければキャリーをセットしてリタ
9BA1 FE50             CPI  50H       ーン
9BA3 CABD9B           JZ   Q5
9BA6 FE42      Q3     CPI  42H
9BA8 C2AD9B           JNZ  Q6
9BAB AF               XRA  A
9BAC C9               RET
9BAD FE44      Q6     CPI  44H
9BAF C2B59B           JNZ  Q7
9BB2 3E10             MVI  A,10H
9BB4 C9               RET
9BB5 FE48      Q7     CPI  48H
9BB7 C2E29B           JNZ  A9
9BBA 3E20             MVI  A,20H
9BBC C9               RET
9BBD 1A        Q5     LDAX D
9BBE 13               INX  D
9BBF FE53             CPI  53H
9BC1 C2E29B           JNZ  A9
9BC4 1A               LDAX D
9BC5 13               INX  D
9BC6 FE57             CPI  57H
9BC8 C3969B           JMP  Q4
9BCB CDCEED   CHEC3   CALL EDCEH     〔DE〕がB，C，D，E，H，L，
9BCE C5               PUSH B         M，Aのどれかか調べる。どれで
9BCF E5               PUSH H         もなければキャリーをセットして
9BD0 21B09D           LXI  H,BFBC    リターン
9BD3 0608             MVI  B,08H
9BD5 1A               LDAX D
9BD6 13               INX  D
9BD7 BE        Q8     CMP  M
9BD8 CAE69B           JZ   C3
9BDB 23               INX  H
9BDC 05               DCR  B
9BDD C2D79B           JNZ  Q8
9BE0 E1               POP  H
9BE1 E1               POP  H
9BE2 E1        A9     POP  H
9BE3 C3B295           JMP  G4
9BE6 3E08      C3     MVI  A,08H
9BE8 90               SUB  B
9BE9 E1               POP  H
9BEA C1               POP  B
9BEB C9               RET
9BEC CDCEED   CLCH    CALL EDCEH     〔DE〕から先が0D，0Aで終わ
9BEF C3A6DE           JMP  DEA6H     っているか調べる
9BF2 CDCEED   CHECC   CALL EDCEH     〔DE〕が「，」か調べる
9BF5 1A               LDAX D
9BF6 13               INX  D
9BF7 FE2C             CPI  2CH
9BF9 C8               RZ
9BFA 37               STC
9BFB C9               RET
9BFC 00               NOP
9BFD 00               NOP
9BFE 00               NOP
9BFF 00               NOP
```

メッセージバッファ

```
9C00 BFISR  0B 4E 4F 50 BB 4C 58 49 EC 53 54 41 58 AB 49 4E
9C10        58 2B 49 4E 52 2B 44 43 52 33 4D 56 49 0B 52 4C
9C20        43 00 AB 44 41 44 EC 4C 44 41 58 AB 44 43 58 28
9C30        28 30 0B 52 52 43 00 B8 E8 A8 28 28 30 0B 52 41
9C40        4C 00 A8 E8 A8 28 28 30 0B 52 41 52 00 B8 1C 53
9C50        48 4C 44 A8 28 28 30 0B 44 41 41 00 A8 1C 4C 48
9C60        4C 44 A8 28 28 30 0B 43 4D 41 00 B8 1B 53 54 41
9C70        A8 28 28 30 0B 53 54 43 00 A8 1B 4C 44 41 A8 28
9C80        28 30 0B 43 4D 43 8B 4D 4F 56 6B 41 44 44 6B 41
9C90        44 43 6B 53 55 42 6B 53 42 42 6B 41 4E 41 6B 58
9CA0        52 41 6B 4F 52 41 6B 43 4D 50 0B 52 4E 5A CB 50
9CB0        4F 50 1B 4A 4E 5A 1B 4A 4D 50 1B 43 4E 5A CC 50
9CC0        55 53 48 13 41 44 49 4B 52 53 54 0A 52 5A 0B 52
9CD0        45 54 1A 4A 5A 00 1A 43 5A 1C 43 41 4C 4C 13 41
9CE0        43 49 48 0B 52 4E 43 C8 1B 4A 4E 43 13 4F 55 54
9CF0        1B 43 4E 43 C8 13 53 55 49 48 0A 52 43 00 1A 4A
9D00        43 12 49 4E 1A 43 43 00 13 53 42 49 48 0B 52 50
9D10        4F C8 1B 4A 50 4F 0C 58 54 48 4C 1B 43 50 4F C8
9D20        13 41 4E 49 48 0B 52 50 45 0C 50 43 48 4C 1B 4A
9D30        50 45 0C 58 43 48 47 1B 43 50 45 00 13 58 52 49
9D40        48 0A 52 50 C8 1A 4A 50 0A 44 49 1A 43 50 C8 13
9D50        4F 52 49 48 0A 52 4D 0C 53 50 48 4C 1A 4A 4D 0A
9D60        45 49 1A 43 4D 00 13 43 50 49 48 0B 48 4C 54
9D6F BFTTL  19 2A 4C 45 56 45 4C 2D 32 20 41 53 53 45 4D 42
9D7F        4C 45 52 20 56 31 2E 30 2A 0D
9D89 BFLO   0C 0D D7 CD DE D9 20 B5 B0 CA DE B0 0D
9D96 BFST   08 53 54 4F 50 20 41 54 20
9D9F BFNA   10 0D 4E 4F 20 41 44 44 52 45 53 53 20 4F 46 20
9DAF        22
9DB0 BFBC   42 43 44 45 48 4C 4D 41
9DB8 BFSP   01 42 01 44 01 48 02 53 50
9DC1 BFPSW  01 42 01 44 01 48 03 50 53 57
9DCB BFEQU  05 20 45 51 55 20
9DD1 BFLB   08 2D 2D 4C 41 42 45 4C 20
9DDA BFIN   06 49 4E 50 55 54 0D
9DE1 BFDL   07 44 45 4C 45 54 45 0D
9DE9 BFNP   0C 20 4E 4F 20 50 52 4F 47 52 41 4D 0D
9DF6 BF-    05 2D 2D 2D 2D
9DFB BFBL5  20 20 20 20 20
9E00 BF*    2A 2A 2A
9E03 BFCOM  4F 52 47 80 90 44 53 41 BD 90 43 4F 4E 8D 91 4C
9E13        41 42 98 91 50 52 47 FC 91 43 4C 52 4C 92 48 44
9E23        53 55 92 44 45 4C 9A 92 4E 41 4C 18 93 4E 45 57
9E33        85 93 4D 4F 56 A4 93 00
```

拡張コマンドCAL, LCT

追加するコマンドは，ＣＡＬとＬＣＴです。

ＣＡＬは，ＢＡＳＩＣモードのＣＡＬＬと同じですが，モニタのＧＯコマンド同様にも使えます。今までは作ったプログラムにジャンプするのにＢＲＥＡＫキーを押してＧＯ，××××としなければならなかったのが，これで直接ジャンプするようにできます。また，サブルーチンがちゃんと働くかどうかを調べるのにも便利です（実行がおわればもとにもどってきますから）。

たとえば，ＢＳＤ－１２００ＭＴ（ＢＳ専用のオートカセットデッキ）を動かすのにＢＡＳＩＣでＣＡＬＬ５０３Ｈとしますが，ここから直接ＣＡＬ，５０３とすれば同様に使えます。

ＬＣＴは，今入っているラベルの個数を出すコマンドです。ＬＣＴ〔復改〕と入力すれば，ラベルの個数が10進数で表示されます。

せっかくこれがＡＳＳＥＭＢＬＥＲなのに機械語で入力するなんてばかばかしいので，プログラムに機械語は付けてありません。ですからＡＳＳＥＭＢＬＥＲをスタートさせた後，**第2-5表**にしたがって入力して下さい。また，ＬＣＴ，ＣＡＬの省略形は一応あってそれぞれＬＣ．，ＣＡ．ですが，これでは省略しなくても同じです。

∧：入力の訂正コマンド

ＯＲＧコマンドで入力していて，たとえばＭＶＩＡ，16Ｈと入力するところをＭＶＩＡ，17Ｈと入力してしまったときは，まだ〔復改〕を押さないうちなら〔後退〕キーで直せますが，〔復改〕を押してしまった後だと再びＯＲＧ，××××と入力しなければならなく，大変不便です。そこで，まちがえた時等戻れるようにしたのがこの機能です。

まちがえて入力した時は，「∧」（べき乗のマーク），〔復改〕と入力すれば画面は一行戻って，もう一度入力できるようになります。

この機能は，一行だけでなく好きなだけバックできます。

《第2－5表》LCT, CALコマンド

```
NEW 1)
ORG, 9E3A
"CAL"
+60, 9E
"LCT"
+72, 9E, 0
復改 2)
ORG, 9E60
CALL 9B58H
JNZ  9473H
CALL 9BECH
JNZ  9473H
PUSH H
LXI  H, 94A6H
XTHL
PCHL
JNZ  946DH 3)
LDA  8BDDH
MOV  L, A
MVI  H, 0
MVI  C, 4
CALL ED5FH
CALL ED92H
JMP  9026H
復改
```

入力するときの注意

[1] NEWは，必ずしも入力する必要はありません。

[2] このあと，コマンド待ちになりますが，CALとLCTはまだ使わないで下さい（使うと暴走します）。

[3] これを入力したアドレスが9E72であることを確かめて下さい。

※各行の最後には，もちろん〔復改〕を入力して下さい。

ＨＬＢ

プログラムをプリンターに打ち出すＨＤＳというコマンドがありますが，ラベルも打ち出せたほうが便利なので，ＨＬＢというコマンドを追加しました。ＨＬＢもＨＤＳと同様，１文字ごとにプログラムの実行が84ＥＣに出てきます。本文中でも述べたように，ＢＡＳＩＣのＨＬＳＴと同様にすればいいのです。

ＬＥＶＥＬ２アセンブラは，ＨＤＳコマンド実行中，「！」で止められます。ＬＡＢが「！」で止められるのと同様，ＨＬＢも「！」で止められます。また省略形はＨＬ．です。

《第2－6表》∧コマンドプログラム・リスト

```
NEW                    *9478：JMP  94A6H
ORG, 9E86              *9409：JNZ  946DH  92
JZ 958FH               *920F：JNZ  946DH
CPI 5EH                 9E15：＋C3, 9E
JNZ 94D1H               919B：LXI H, 9ECEH
INX D                   91AE：＋C8, 0, 0
CALL 9BECH              91B7：＋D8, 0, 0
JNZ 95CCH               ORG, 9EBB
LXI H, 0000H            JNZ 946DH
DAD SP                  MVI A, 55H
SHLD 8BFBH              STA 862CH
LHLD 8BE6H              CALL 9198H
LXI D, FFF0H            MVI A, 56H
XCHG                    STA 862CH
DAD D                   JMP 9026H
PUSH H                  CALL 91AAH
CALL 9988H              JMP 9026H
MVI B, 00H              〔復改〕
MOV C, A                ORG, 9E44
DAD B                   "HLB"
CALL EE68H              ＋BB, 9E, 0
JM 9EA4H                〔復改〕
POP H
SHLD 8BE6H
LHLD 8BFBH
SPHL
RET
〔復改〕
94CE：JMP 9E86H 1)
```

1) ここまででORGで「∧」が使えるようになります。次からはもう使ってもいいです。

プログラムの入力

アセンブラで**第2-6表**にしたがって入力して下さい。ここで＊をつけた3行は今回の拡張にはあまり関係ありません。(でも絶対に入力して下さい。入力しないとＬＡＢ，ＨＬＢを使った時におかしくなります。)

もう気がついた人もいると思いますが，ＰＲＧコマンドで，ＰＲＧ，××××のあとはでたらめなものを入力してもエラーメッセージがでないようになっています。これでは困るので，それを訂正しただけのことです。

LEVEL 2 アセンブラDe-bug

ＬＥＶＥＬ２アセンブラのＤＥＬコマンドにｂｕｇがありました。プログラムを作ってから気がつきましたので訂正してください。

このｂｕｇは直した方が少しプログラムが短くなったのですが，プログラム領域はふえません。（そのかわりＮＯＰが入りみっともないプログラムになりますが）。

ｄｅ－ｂｕｇするためには，**第2－7表**のとおり入力して下さい。

《第2－7表》デバッグのためのプログラム

```
〔ＢＲＥＡＫ〕……モニタへ
ＭＭ，９２ＡＣ，９２Ｄ５，９２ＡＢ
〔復改〕……………アセンブラにもどる
９２Ａ９：＋Ｅ５，００
９２ＢＦ：＋Ｅ１，Ｅ５，００，ＣＤ，７Ｆ，
            Ｆ７，ＤＡ，７０，９４，ＣＤ，
            ８８，Ｆ７，２Ｂ
９２Ｄ５：＋Ｄ１
```

最後にプログラムをＲＯＭ化する場合や，長いプログラム開発に便利なようにプログラムをＣ０00に移すプログラムを**第2－8表**に載せておきます。

《第2－8表》Ｃ０００へのプログラム移動

```
ＮＥＷ
ＰＲＧ，９０００，Ａ０００
ＭＯＶ，９０００，Ａ０００，Ｃ０００
ＣＥ０７：＋Ｃ０
ＣＥ０Ｃ：＋Ｃ０
ＣＥ１１：＋Ｃ１
ＣＥ１６：＋ＣＥ
ＣＥ１Ｂ：＋Ｃ１
ＣＥ２０：＋Ｃ２
ＣＥ２５：＋Ｃ２
ＣＥ２Ａ：＋Ｃ２
ＣＥ２Ｆ：＋Ｃ３
ＣＥ３４：＋Ｃ３
ＣＥ３９：＋Ｃ３
ＣＥ３Ｅ：＋ＣＥ
ＣＥ４３：＋ＣＥ
ＣＥ４８：＋ＣＥ
```

3

LKIT-16機能拡張

プログラム・パッケージ

LKIT-16　　樺　茂

LKIT16の3K語BASICはカラーグラフィック機能を有する浮動小数点形式のBASICで，ゲーム等，ディスプレイ等には使い易いと思います。しかしながらANSI-minimal BASICに準拠した3K語（6KB）BASICではfiling等の実用プログラム，或いは高級ゲーム等には物足りず，各種の命令を機械語で作成しCALL文で呼び出す必要があります。

その点ではアセンブラ専用機として開発された本機の特性を活かしてユーザー作成の機械語サブプログラムを利用し易いように配慮されています。私は大型計算機で使用する変数名の整理等の実用面で本機を利用するためサブルーチンをいくつか組んでみました。

今回他機種のレベルIIクラスのBASIC機能を目標として各種のプログラムをまとめてパッケージにしてみました。LKIT16の持つ多数のキーを利用するリアルタイム処理プログラム（AKBKサブルーチン）を既に月刊マイコン'79年2月号に発表しましたが，これももちろん入っています。未熟なプログラムで，エラー検出機能も乏しいものですが発表させて頂きます。

更に良いものが発表されることを願っております。

BASICインタプリター内のサブルーチンが利用出来ればかなり短縮できるかと思いますが，インタプリタのプログラムが公開されていないので，マニュアルに掲載されているサブルーチン以外は引用してありません。

プログラムは拡張メモリボード上のX'1FFF'番地迄RAM実装として最後部に格納するようにしてあります。パターン書きこみプログラム(P—1)を使用しないならばX'1E80'～X'1FFC'番地の格納，更にEdit機能拡張のみならばX'1F30～X'1FFB'番地の格納となります。パターン書きこみプログラムをフルに活用するためには，X'1C00'番地以降を使用することになります。理想的には，常用漢字パターン，英小文字パターン，グラフィックパターン等のデータ（**第3—5表**参照），プログラムで丁度1K語になりますので増設メモリーのROMに入れると使い易いと思います。

プログラムの内容は表にまとめましたが，若干の補足を以下にします。

1）プログラム格納領域

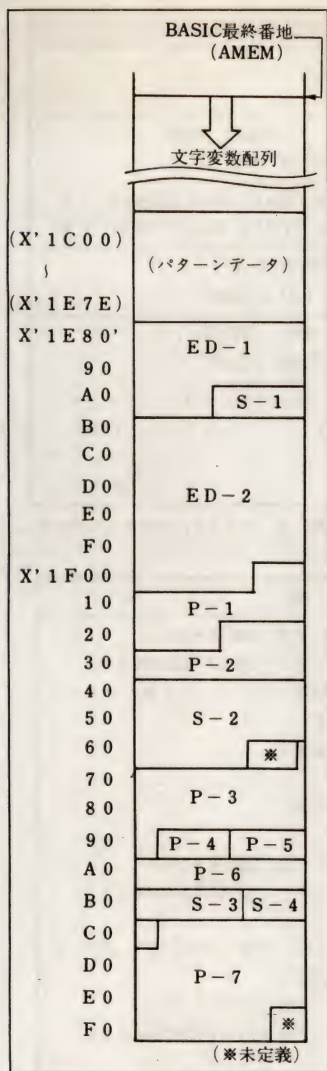
（第3—1図，第3—1表）

X'1E80'1FFF'番地に格納します。プログラムリスト内のカッコで囲まれた機械語は格納領域によって書き変える必要があります。

2）Edit機能拡張のためのプログラム

（第3—2表）

機能および使用法を**第3—2表**に示します。リナンバーは行番号を10番毎につけ直します。いずれ

《第3−1図》
メモリマップ上のプログラム

エリア	格納プログラム	格納番地
ED−1	BLDEL(ブロック消去)	X'1E80'−X'1EA6'
S−1	サブルーチンアドレステーブル	X'1EA7'−X'1EAF'
ED−2	S&R, RENO	X'1EB0'−X'1F0B'
P−1	PATTWT	X'1F0E'−X'1F28'
P−2	STDIM(W), STDIM(R)	X'1F29'−X'1F3F'
S−2	サブルーチン群	X'1F40'−X'1F6A'
P−3	LEFT$, MID$	X'1F6F'−X'1F91'
P−4	PEEK	X'1F92'−X'1F98'
P−5	POKE	X'1F99'−X'1F9F'
P−6	LEN, ASC$	X'1FA0'−X'1FAF'
S−3	サブルーチン(BTOF(I))	X'1FB0'−X'1FB9'
S−4	サブルーチン(WE)	X'1FBA'−X'1FC1'
P−7	(BKCHECK, BKWAIT AKCHECK, AKWAIT)	X'1FC2'−X'1FFC'

《第3−1表》各プログラムの格納領域

プログラム	略号	機能	使用法	結果
行ブロック消去 (ED−1)	BLDEL	BASICプログラム 行番号 V0−nnnn ∫ V1−mmmm を消去する	BASIC直接実行文 :LIST :V0=nnnn:V1=mmmm(註) リセット 機械語プログラム X'1E80'番地よりRUN	BASIC 再起動 :LIST 確認
リナンバー (ED−2)	RENO	BASICプログラム 全行番号を10から 10番毎の数字に おきかえる	BASIC直接実行文 :LIST リセット 機械語プログラム X'1EB9'番地よりRUN	BASIC 再起動 :LIST 確認
シフト & リナンバー (ED−2)	S&R	BASICプログラムの 行番号V0−nnnn 以降をV1−mmmm 以降に移し リナンバーする	BASIC直接実行文 :LIST :V0=nnnn:V1=mmmm(註) リセット 機械語プログラム X'1EB0'番地より	BASIC 再起動 :LIST 確認

(註, V0, V1は必ず1文多行の直接実行文とする)

《第3−2表》EDIT機能拡張のためのプログラム

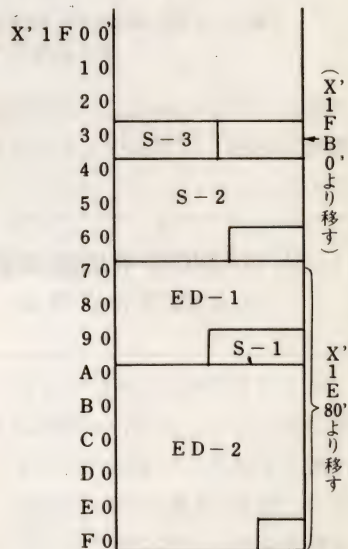
もリセット後機械語プログラムで実行し，BASIC再起動状態に戻ります。

メモリーを節約しEdit機能のみ使用する場合には**第3−2図**の如く，S-3 (BTOF) をX'1F30'番地に移します。その際にS-1内のX'1EAD'番地のX'1FB0'をX'1F30'と訂正して下さい。**第3−2図**の如く格納した場合のプログラム開始番地は，

BLDEL-X'1F70'
RENO-X'1FA9'
S&R-X'1FA0'

となります。

なお，RENO，S&Rでステップを2番毎，5番毎

《第3−2図》
ED-1，ED-2のみ使用の場合

プログラム	略号	先頭番地	BASICプログラム内での使用法
基本パターンデータ書きこみ	STPATTWT	X'1F10'	:ℓℓℓℓ CALL 7952
パターンデータ書きこみ	PATTWT	X'1F13'	:ℓℓℓℓ V0=nnnn：CALL 7955
文字列変数配列(書きこみ)	STDIM(W)	X'1F35'	:ℓℓℓℓ V0=n：X$="○○○…"：CALL 7989
〃 (読み出し)	STDIM(R)	X'1F39'	:ℓℓℓℓ V0=n：CALL 7993：PRINT X$
LEFT$	LEFT$	X'1F70'	:ℓℓℓℓ X$="○○…"：V0=n：CALL 8048：PRINT Y$
MID $	MID $	X'1F70'	:ℓℓℓℓ X$="○○…"：V0=n：CALL 8048：PRINT Z$
メモリー内データ読み出し	PEEK	X'1F92'	:ℓℓℓℓ V0=nnnn：CALL 8082：PRINT V1
〃 書きこみ	POKE	X'1F99'	:ℓℓℓℓ V0=nnnn：V1=mmmm：CALL 8089
文字列変数語長	LEN	X'1FA0'	:ℓℓℓℓ X$="○○…"：CALL 8096：PRINT V0
〃 先頭文字アスキーコード	ASC$	X'1FA9'	:ℓℓℓℓ X$="○○…"：CALL 8105：PRINT V0
キー入力リアルタイム処理	BKCHECK	X'1FC2'	:ℓℓℓℓ CALL 8130(フルキーボード及びテンキーのキー入力チェック)
〃	BKWAIT	X'1FCE'	:ℓℓℓℓ CALL 8142(〃 キー入力待ち)
〃	AKCHECK	X'1FDB'	:ℓℓℓℓ CALL 8155(アセンブラキー キー入力チェック)
〃	AKWAIT	X'1FE6'	:ℓℓℓℓ CALL 8166(〃 キー入力待ち)

(ℓℓℓℓ—行番地、n、nnnn，mmmm—10進数)

《第3—3表》BASIC命令拡張のためのサブルーチン

プログラム略号	V0	V1	V2	X$	Y$	Z$
STPATTWT						
PATTWT	◎					
STDIM(W)	◎	○	○	◎		
STDIM(R)	◎	○	○	◎		
LEFT$	◎	○	○	◎	◎	
MID$	◎	○	○	◎	○	◎
PEEK	◎	◎				
POKE	◎	◎				
LEN	◎	○	○	◎		
ASC$	◎	○	○	◎		
BKCHECK	◎					
BKWAIT	◎					
AKCHECK	◎	◎	◎			
AKWAIT	◎	◎	◎			

(◎使用する変数，○使用しないが定義のみ必要とする変数)
(必ず表の順番に定義すること，必ずプログラムの最初に定義すること)
例）：10␣V0=0：V1=0：V2=0：X$="␣"：Y$="␣"：Z$="␣"

《第3—4表》BASIC〜機械語の受け渡しに使用する変数

番地	機械語		内容
1C00	012C	先頭番地	出力データ数(30画素＊10)₁₀
1C01	0080		パターンコード(PATTERN96より)
1C02 〜 1C97			高分解能パターン，漢字等のパターンデータ（30画素分）
1C98 〜 1C9B			（未定義）
1C9C	0140	先頭番地	出力データ数(32画素＊10)₁₀
1C9D	0080		パターンコード(PATTERN96より)
1C9E 〜 1D3D			英小文字，特殊文字のパターンデータ（32画素分）
1D3E	0280	先頭番地	出力データ数(64画素＊10)₁₀
1D3F	0080		パターンコード(PATTERN96より)
1D40 〜 1E7F			基本パターンデータ（64画素分）グラフィック，図表用

《第3—5表》パターンデータ書込みプログラム用パターンデータメモリマップ

等細かくするためには，1EB7番地(400A)，1EF8番地（480A）の下位バイトにそれぞれ'02'又は'05'を入れて下さい。

3）BASIC命令機能拡張のためのプログラム

すべてBASICプログラム実行中にCALL文で呼び出します。そのため機械語との間に変数の受け渡しを必要としますので必ずプログラムの一番最初に**第3—4表**の変数を定義して下さい。変数名は数値変数，文字変数の区別が守られ，順に定義されていれば何を使っても構いません。

（P-1）パターンデータ書きこみ

序文で述べたようにこのプログラムをフルに活用するためにはX'1E80'番地以前にパターンデータを必要とするため大量のメモリーを要します。

パターンデータ先頭にデータ数（パターン数＊10)₁₀，次番地にX'0080'(パターン96から始まる場合）或いはX'00A0'（パターン224から始まる場合）等先頭パターンコードを入れます。次いでパターンデータを順に1パターンにつき5語（文献1参照）ずつ入れます。BASICプログラム内でV0にパタ

ーンデータ先頭番地(10進数)を入れてからCALL 7955で書きこみます。

例として**第3—5表**に基本パターン(64画素)，英小文字(32画素)，漢字(30画素)のデータを書きこむ場合のアドレスを示します。この場合，X'1C00'～X'1FFF'で丁度1K語となります。

X'1F0F'番地に基本パターン64画素のデータ先頭番地X'1D3E'が入っているとすればBASICプログラムでCALL 7952によって瞬時に書きこみをします。英小文字の場合はV0=7324：CALL 7955，漢字の場合にはV0=7768：CALL 7955で書きこむわけです。

TVIFOPボードのROMにパターンが書きこんであれば通常の使用には十分ですがこのルーチンによればプログラム中で自由自在にパターンを選ぶことが出来ますし，必要なパターンをPATTERN文で追加することも自由です。

(P-2) 文字列変数の配列

第3—3表の列ではX$ (n) に相当する文字列変数を書きこみ[STDIM (W)]，読み出し[STDIM (R)] しています。

配列のデータを**第3—1図**の如くMAP指定の次の番地以降に入れますので，機械語格納領域に喰いこまぬよう計算して使用して下さい。文字列変数1語についてX'A'語のメモリーを必要とします。エラー検出をしませんので配列の大きい数を指定すると機械語を破壊します。

(P-3) LEFT$，MID$相当 サブルーチン

X$の文字列の内，V0に指定された数字迄の文字をY$に，V0番目の文字をZ$に入れます。RIGT$は，次のLENとMID$を組み合わせて作成可能です。

(P-4，5) PEEK，POKE相当 サブルーチン

V0で示した番地(10進数)のデータV1(10進数)の書き込み，読み出しをします。但しX'7FFF'迄は正数ですがX'8000'～X'FFFF'は負数になります。注意して下さい。

(P-6) LEN，ASC$相当 サブルーチン

X$の文字数(LEN)，第1字のASCIIコード(ASC$) をV0に入れます。

(P-7) キー入力リアルタイム処理

既に「月刊マイコン'79年」2月号に掲載されたプログラムですが，パッケージ収納に際して一部変更しました。内容はほぼ既述のものと同様です。尚，AK処理に際してV1，V2に列，行のデータが入るようになりました。但し，行のデータは2倍した数になっています。呼び出し番地も変更してあります。

4) プログラム格納領域の変更について

メモリーの他の領域に移す場合にはリスト内のカッコで囲んだ語を変更する必要があります。

X'1EA7'～X'1EAD'

X'1F0E'～X'1F0F'

を移動分だけ加減して下さい。

X'1F0E'は基本パターンデータ先頭番地ですから格納番地によって書きかえて下さい。

5) おわりに

素人が手当り次第に組んだプログラムですが，何とか動いて役に立っています。エラー検出機能が充分でないので慎重に使用して下さい。随分無駄があるようですのでもっとすっきりしたプログラムを発表して頂ければ幸いです。

基本図形パターン，小文字パターン常用漢字パターン等の書きこみについても機会があれば供覧したいと考えています。

何時の日にかパナファコムから拡張BASICが発表されることを願っています。

(プログラム作成に際して，名古屋マイコンセンターの木村・野原，名古屋バイトショップの高田，各氏のアドバイスを頂きました。感謝致します。)

〈参考文献〉

1) LKIT16 入出力ルーチン使用説明書
2) LKIT16 BASIC説明書

《第3—6表》プログラムリスト

No.	番地	機械語	ラベル	アセンブラ語		注釈
ED-1	1E80	6000	BLDEL	CLEAR,	R0	
	1	8043		ST	R0,X'43'	
	2	9F28		BAL	(*+28)	⇄VIN4041
	3	C040		L	R0,X'40'	
	4	C141		L	R1,X'41'	
	5	5138		C	R1,R0,PZ	
	6	CF26		B	(*+26)	→E-OF
	7	9F20	VOCHECK	BAL	(*+20)	⇄LNOSORT
	8	7A4A		SKIP	R2,Z	
	9	CF05		B	*+5	→V1CHECK
	A	C043		L	R0,X'43'	
	B	7848		SKIP	R0,Z	
	C	CF18		B	*+18	→EPOG修正
	D	CF19		B	*+19	→FINISH
	E	5269	V1CHECK	C	R2,R1,MZ	
	F	CF09		B	*+9	→OVER
	1E90	C243	BETWN	L	R2,X'43'	
	1	7A5A		SKIP	R2,NZ	
	2	8342		ST	X0,X'42'	
	3	E400		L	X1,X'0'(X0)	
	4	5A0C		A	R2,X1	
	5	8243		ST	R2,X'43'	
	6	9F12		BAL	(*+12)	⇄NEXTLINE
	7	CFF1		B	*-F	
	8	C043	OVER	L	R0,X'43'	
	9	7858		SKIP	R0,NZ	
	A	CF0C		B	*+C	
	B	C442		L	X1,X'42'	
	C	535C		C	X0,X1,NZ	
	D	CF09		B	*+9	→FINISH
	E	7808		NOP		
	F	C290		L	R2,X'9D'	
	1EA0	5A03		S	R2,X0	
	1	9F08		BAL	(*+8)	⇄TRANSP
	2	C49D		L	X1,X'9D	
	3	C043		L	R0,X'43'	
	4	5C00	EPOG修正	S	X1,R0	
	5	849D		ST	X1,X'9D'	
	6	DF08	FINISH	B	(*+8)	→BASIC
S-1	7	(1F40)	アドレス表	DC		LHOSORT
	8	(1F44)		DC		NEXTLINE
	9	(1F4F)		DC		TRANSP
	A	(1F56)		DC		VIN4041
	B	(1F5B)		DC		VOIN40
	C	(1F68)		DC		E-OF
	D	(1FB0)		DC		BTOF(I)
	E	D801		DC		BASIC(再)

No.	番地	機械語	ラベル	アセンブラ語		注釈
S-1	1EAF	——			(未定義)	
ED-2	1EB0	9FFA	S&R	BAL	(*-6)	⇄VIN4041
	1	C040		L	R0,X'40'	
	2	9FF5		BAL	(*-B)	⇄LNOSORT
	3	7A5A		SKIP	R2,NZ	
	4	DFFA		B	(*-6)	→FINISH
	5	8340		ST	X0,X'40'	
	6	C041		L	R0,X'41'	
	7	400A		SI	R0,X'A'	
	8	CF04		B	*+4	→RENOMAIN
	9	C372	RENO	L	X0,X'72'	
	A	8340		ST	X0,X'40'	
	B	6000		CLEAR,	R0	
	C	8041	RENOMAIN	ST	R0,X'41'	
	D	C372		L	X0,X'72'	
	E	7C0B		MV	X1,X0	
	F	E000		L	R0,X'0'(X0)	
	1EC0	5C08		A	X1,R0	
	1	4B02		AI	X0,X'2'	
	2	E000	B命令SORT	L	R0,X'0'(X0)	
	3	CA47		L	R2,*+47	
	4	505A		C	R0,R2,NZ	GOTO?
	5	CF0E		B	*+E	
	6	4A0F		AI	R2,X'F'	
	7	505A		C	R0,R2,NZ	GOSUB?
	8	CF0B		B	*+B	
	9	4A08		AI	R2,X'8'	
	A	505A		C	R0,R2,NZ	THEN?
	B	CF08		B	*+8	
	C	4B01		AI	X0,X'1'	
	D	534C		C	X0,X1,Z	
	E	CFF4		B	*-C	→B命令SORT
	F	C29D		L	R2,X'9D'	
	1ED0	544A		C	X1;R2, Z	
	1	CFED		B	*-13	
	2	CF30		B	*+30	→LNO処理
	3	4B01		AI	X0,X'1'	
	4	8F1E		BAL	*+1E	⇄SUB1
	5	2301		PUSH	X0	
	6	C342		L	X0,X'42'	
	7	C09D		L	R0,X'9D'	
	8	5348		C	X0,R0,Z	
	9	CF03		B	*+3	
	A	2302		POP	X0	
	B	CF11		B	*+11	
	C	8F1B		BAL	*+1B	⇄SUB2
	D	C043		L	R0,X'43'	

No.	番地	ラベル	ラベル	アセンブラ語		注釈
ED-2	1EDE	9FCF		BAL	(*−31)	⇄BTOF
	F	2302		POP	X0	
	1EE0	E201		L	R2,X'1'(X0)	
	1	504A		C	R0,R2,Z	R0=上位？
	2	CFF3		B	*−D	
	3	E202		L	R2,X'2'(X0)	
	4	514A		C	R1,R2,Z	R1=下位？
	5	CFF0		B	*−10	
	6	2301		PUSH	X0	
	7	C044		L	R0,X'44'	
	8	9FC5		BAL	(*−3B)	⇄BTOF
	9	2302		POP	X0	
	A	A001		ST	R0,X'1'(X0)	修正
	B	A102		ST	R1,X'2'(X0)	
	C	4B03		AI	X0,X'3'	
	D	E000		L	R0,X'0'(X0)	(，)あり？
	E	C91D		L	R1,*+1D	ON GOTO処理
	F	5059		C	R0,R1,NZ	
	1EF0	CFE4		B	*−1C	
	1	CFDB		B	*−25	
	2	C040	SUB1	L	R0,X'40'	
	3	8042		ST	R0,X'42'	
	4	C041		L	R0,X'41'	
	5	8044		ST	R0,X'44'	
	6	2003		RET		
	7	C044	SUB2	L	R0,X'44'	
	8	480A		AI	R0,X'A'	
	9	8044		ST	R0,X'44'	
	A	C342		L	X0,X'42'	
	B	E101		L	R1,X'1'(X0)	
	C	8143		ST	R1,X'43'	
	D	7A0B		MV	R2,X0	
	E	E100		L	R1,X'0'(X0)	
	F	590A		A	R1, R2	
	1F00	8142		ST	R1,X'42'	
	1	2003		RET		
	2	8FF0	LNO処理	BAL	*−10	⇄SUB 1
	3	8FF4		BAL	*−C	⇄SUB 2
	4	A001		ST	R0,X'1'(X0)	
	5	C342		L	X0,X'42'	
	6	C09D		L	R0,X'9D'	
	7	5348		C	X0,R0,Z	
	8	CFFB		B	*−5	
	9	DFA5	FINISH	B	(*−5B)	→BASIC
	A	0307		DC		(GOTO)
	B	2C09		DC		(コンマ)
	1F0C	——			(未定義)	
	D	——			(未定義)	
P-1	1F0E	(1FBA)		DC		WECHECK
	F	(1D3E)		DC		基本パターンデータ先頭

No.	番地	機械語	ラベル	アセンブラ語		注釈
P-1	1F10	C8FF	STPATTWT	L	R0,*−1	基本パターン書きこみ
	1	8040		ST	R0,X'40'	
	2	CF02		B	*+2	
	3	8F48	PATTWT	BAL	*+48	VOIN40
	4	C340		L	X0,X'40'	先頭番地よみ出し
	5	E400		L	X1,X'0'(X0)	
	6	080A		MVI	R0,X'A'	パターン書きこみ
	7	1050		WT	R0,X'50'	モード
	8	4B01		AI	X0,X'1'	
	9	E000		L	R0,X'0'(X0)	
	A	9FF4		BAL	(*−C)	パターンコード
	B	1053		WT	R0,X'53'	
	C	4B01		AI	X0,X'1'	パターンデータ書きこみ
	D	E000		L	R0,X'0'(X0)	
	E	7008		BSWP	R0,R0	
	F	9FEF		BAL	(*−11)	
	1F20	1053		WT	R0,X'53'	
	1	4471		SI	X1,X'1',P	データ終了？
	2	CF04		B	*+4	
	3	2C4F		TBIT	X1,X'F',Z	
	4	CFFA		B	*−6	奇数→バイト交換
	5	CFF7		B	*−9	偶数→次語
	6	0808		MVI	R0,X'08'	リフレッシュメモリ
	7	1050		WT	R0,X'50'	ライトモード
	8	2003		RET		→BASIC
P-2	9	8F32	SUB	BAL	*+32	⇄VOIN40
	A	4307		SI	X0,X'7'	
	B	E300		L	X0,X'0'(X0)	
	C	8340		ST	X0,X'40'	
	D	C473		L	X1,X'73'	
	E	4C01		AI	X1,X'1'	
	F	4031		SI	R0,X'1',PZ	
	1F30	CF03		B	*+3	
	1	4C0A		AI	X1,X'A'	
	2	CFFD		B	*−3	
	3	8441		ST	X1,X'41'	
	4	2003		RET		
	5	8FF4	STDIM(W)	BAL	*−1C	
	6	C441		L	X1,X'41'	
	7	C340		L	X0,X'40'	
	8	CF04		B	*+4	
	9	8FF0	STDIM(R)	BAL	*−10	
	A	C341		L	X0,X'41'	
	B	C440		L	X1,X'40'	
	C	6202	転送	CLEAR	R2	
	D	0A0A		MVI	R2,X'A'	
	E	8F11		BAL	*+11	⇄TRANSP(S)
	F	2003		RET		→BASIC
S-2	1F40	C372	LNOSORT	L	X0,X'72'	
	1	E201		L	R2,X'1'(X0)	

No.	番地	機械語	ラベル	アセンブラ語		注釈
S-2	1F42	5228		C	R2,R0,M	
	3	2003		RET		
	4	E200	NEXTLINF	L	R2,X'0'(X0)	
	5	5B0A		A	X0,R2	
	6	C49D		L	X1,X'9D'	END?
	7	534C		C	X0,X1,Z	
	8	CFF9		B	*−7	
	9	6202		CLEAR	R2	
	A	2003		RET		
	B	——		(未定義)		
	C	——		(〃)		
	D	8F02	TRANSP(R)	BAL	*+2	転送ルーチン
	E	972D		H		
	F	E100	TRANSP(S)	L	R1,X'0'(X0)	転送サブルーチン
	1F50	A900		ST	R1,X'0'(X1)	
	1	4B01		AI	X0;X'1'	
	2	4C01		AI	X1,X'1'	
	3	4241		SI	R2,X'1',Z	
	4	CFFB		B	*−5	
	5	2003		RET		
	6	8F05	VIN4041	BAL	*+5	⇄VOIN40
	7	4303	V1IN41	SI	X0,X'3'	
	8	8F07		BAL	*+7	⇄VFIN
	9	8041		ST	R0,X'41'	
	A	2003		RET		
	B	8F03	VOIN40	BAL	*+3	⇄VOIN
	C	8040		ST	R0,X'40'	
	D	2003		RET		
	E	C374	VOIN	L	X0,X'74'	
	F	E001	VFIN	L	R0,X'1'(X0)	
	1F60	E102		L	R1,X'2'(X0)	
	1	978F	ETOB	BAL	(X'8F')	
	2	9795		BAL	(X'95')	
	3	2003		RET		
	4	080D	E-0D	MVI	R0,X'D'	エラー表示
	5	CF04		B	*+4	
	6	080E	E-0E	MVI	R0,X'E'	
	7	CF03		B	*+3	
	8	080F	E-0F	MVI	R0,X'F'	
	9	9732		BAL	(X'32')	
	A	972D		H		
	B	——		(未定義)		
	C	——				
	D	——				
	E	——				
P-3	F	0020		DC		
	1F70	8FEB	LEFT$,MID$	BAL	*−15	⇄VOIN40
	1	7A08		MV	R2,R0	
	2	4307		SI	X0,X'7'	X0←X$のLOC
	3	E300		L	X0,X'0'(X0)	

No.	番地	機械語	ラベル	アセンブラ語		注釈
P-3	1F74	E100		L	R1,X'0'(X0)	語数オーバー？ →E-0F
	5	5132		CB	R1,R2,PZ	
	6	CFF2		B	*−E	
	7	7C0B		MV	X1,X0	
	8	4C0A		AI	X1,X'A'	X1←Y$のLOC
	9	E800		L	R0,X'0'(X1)	
	A	7802		MVB	R0,R2	
	B	A800		ST	R0,X'0'(X1)	
	C	4C0A		AI	X1,X'A'	Z$−1語
	D	0801		MVI	R0,X'1'	
	E	A800		ST	R0,X'0'(X1)	
	F	2401		push	X1	
	1F80	440A		SI	X1,X'A'	
	1	C9EE		L	R1,*−12	R1=X'0020'
	2	4B01		AI	X0,X'1'	X$の第1語
	3	4C01		AI	X1,X'1'	Y$の第1語
	4	E000		L	R0,X'0'(X0)	
	5	4271		SI	R2,X'1',P	last one ?
	6	CF07		B	*+7	
	7	A800		ST	R0,X'0'(X1)	
	8	4261		ST	R2,X'1',MZ	
	9	CFF9		B	*−7	
	A	7008		BSWP	R0,R0	
	B	7801		MVB	R0,R1	
	C	CF03		B	*+3	
	D	7801	last one	MVB	R0,R1	
	E	A800		ST	R0,X'0'(X1)	
	F	2402		POP	X1	MID$=Z$
	1F90	A801		ST	R0,X'1'(X1)	
	1	2003		RET		→BASIC
P-4	2	8FC9	PEEK	BAL	*−37	⇄VOIN40
	3	7B08		MV	X0,R0	
	4	E000		L	R0,X'0'(X0)	
	5	8F1B		BAL	*+1D	⇄BTOF(I)
	6	C474		L	X1,X'74'	
	7	4403		SI	X1,X'3'	
	8	CF1F		B	*+1F	→VFOUT, BASIC
P-5	9	8FC2	POKE	BAL	*−3E	⇄VOIN40
	A	2001		push	R0	
	B	4303		SI	X0,X,3'	
	C	8FC3		BAL	*−3D	⇄VFIN
	D	2302		POP	X0	
	E	A000		ST	R0,X'0'(X0)	
	F	2003		RET		→BASIC
P-6	1FA0	8F03	LEN	BAL	*+3	⇄READX$
	1	E000		L	R0,X'0'(X0)	
	2	CF0A		B	*+A	
	3	6202	READX$	CLEAR	R2	
	4	0AFF		MVI	R2,X'FF'	
	5	C374		L	X0,X'74'	

No.	番地	機械語	ラベル	アセンブラ語		注釈
P-6	1FA6	4307		SI	X0,X'7'	
	7	E300		L	X0,X'0'(X0)	
	8	2003		RET		
	9	8FFA	ASC$	BAL	*-6	⇄READ X$
	A	E001		L	R0,X'1'(X0)	
	B	7008		BSWP	R0,R0	
	C	680A		AND	R0,R2	
	D	8F03		BAL	*+3	⇄BTOF(I)
	E	8F08		BAL	*+8	⇄VOOUT
	F	2003		RET		→BASIC
S-3	1FB0	6101	BTOF(I)	CLEAR	R1	
	1	6202		CLEAR	R2	
	2	0A0F		MVI	R2,X'F'	
	3	9788		BAL	(X'88')	
	4	7902		MVB	R1,R2	
	5	2003		RET		
	6	C474	VOOUT	L	X1,X'74'	
	7	A801	VFOUT	ST	R0,X'1'(X1)	
	8	A902		ST	R1,X'2'(X1)	
	9	2003		RET		
S-4	A	1A51	WE?	RD	R2,X'51'	
	B	1251		WT	R2,X'51'	
	C	090F		MVI	R1,X'0F'	
	D	6A09		AND	R2,R1	
	E	0902		MVI	R1,X'02'	
	F	5142		CB	R1,R2,Z	
	1FC0	CFFA		B	*-6	
	1	2003		RET		
P-7	2	8FF8	BKCHECK	BAL	*-8	⇄WE?
	3	0800		MVI	R0,X'00'	KBRDモード指定
	4	1050		WT	R0,X'50'	
	5	0A3F		MVI	R2,X'3F'	カウンター指定 X'3FFF'
	6	720A		BSWP	R2, R2	
	7	0AFF		MVI	R2,X'FF'	
	8	4251		SI	R2,X'1',NZ	カウンターX'0'で BASICへ
	9	CF0E		B	*+D	
	A	1851		RD	R0,X'51'	BK入力の有無判定
	B	2859		TBIT	R0,X'9',NZ	
	C	CFFC		B	*-4	
	D	CF07		B	*+7	→RDDATA
	E	8FEC	BKWAIT	BAL	*-14	⇄WE?
	F	0800		MVI	R0,X'00'	KBRDモード指定
	1FD0	1050		WT	R0,X'50'	
	1	1851	RDDATA	RD	R0,X'51'	BK入力の有無判定
	2	2859		TBIT	R0,X'9',NZ	

No.	番地	機械語	ラベル	アセンブラ語		注釈
P-7	1FD3	CFFE		B	*-2	
	4	1852		RD	R0,X'52'	
	5	8FDB		BAL	*-25	⇄BTOF
	6	8FE0		BAL	*-20	⇄VOOUT
	7	8FE3		BAL	*-1D	⇄WE?
	8	0A08		MVI	R1,X'08'	リフレッシュメモリ ライトモード
	9	1250		WT	R1,X'50'	
	A	2003		RET		→BASIC
	B	0A01	AKCHECK	MVI	R2,X'01'	カウンターの設定 X'01FF'
	C	720A		BSWP	R2,R2	
	D	0AFF		MVI	R2,X'FF'	
	E	2201		push	R2	
	F	9733		BAL	(X'33')	⇄KSCAN
	1FE0	2202		POP	R2	
	1	4251		SI	R2,X'1',NZ	カウンターX'0'で BASICへ
	2	2003		RET		
	3	7959		SKIP	R1,NZ	AKの入力有無 判定
	4	CFFA		B	*-6	
	5	CF04		B	*+3	→MAKEVS
	6	9733	AKWAIT	BAL	(X'33)	⇄KSCAN
	7	7959		SKIP	R1,NZ	AKの入力有無判定
	8	CFFE		B	*-2	
	9	8040	MAKEVS	ST	R0,X'40'	
	A	8141		ST	R1,X'41'	
	B	0AFF		MVI	R2,X'FF'	行列のデータを 1変数にまとめる →V0へ
	C	680A		AND	R0,R2	
	D	690A		AND	R1,R2	
	E	210D		SL	R1,RE	
	F	210D		SL	R1,RE	
	1FF0	5809		A	R0,R1	
	1	4007		SI	R0,X'7'	キーデータ調整
	2	8FBE		BAL	*-42	⇄BTOF
	3	8FC3		BAL	*-3D	⇄VOOUT
	4	C040		L	R0,X'40'	列データ→V1へ
	5	8F02		BAL	*+2	
	6	C041		L	R0,X'41'	行データ→V2へ
	7	0AFF		MVI	R2,X'FF'	
	8	680A		AND	R0,R2	
	9	8FB7		BAL	*-49	⇄BTOF
	A	4403		SI	X1,X'3'	
	B	8FBC		BAL	*-44	⇄VFOUT
	C	2003		RET		
	D	—				(未定義)
	E	—				
	F	—				

4 EX-80機能アップ!

M.L.Editorプログラム

EX-80

金川　清

東芝1ボードマイコンＥＸ８０を使い，機械語のプログラムを作る為の編集用プログラムを作りましたので紹介します。約2Kバイトのプログラムですが，ＰＲＯＭ化して 800－ＦＦＦにセットする事によりＥＸ８０本体ボードだけで使用できます。内容はプログラムの転送，Verify，転送先でのアドレスのリロケート，ジャンプ先等の絶対アドレス指定の相対アドレス化及びラベル方式のアドレス化，さらに指定ワードの書込みとVerifyです。

このMahine Language Editor のワーキングRAMエリアは被編集プログラムエリアの他に，パラメータ設定用に83A0～83B3，8400～85FFを使いますので，ＥＸ８０のRAMは8200～87FFの実装が必要です。このままでＥＸ８０ＢＳででも使用できます。

プログラムの考え方

Move，Verify，Write は簡単ですので説明を省略しますが，Verifyでエラーを発見すると，最初のエラーのアドレスとエラー内容，正しい内容を表示してキー入力待ちとなります。

Address Changeは，ジャンプ命令等の３バイト命令のジャンプ先２バイトを自動的に変更します。指定したアドレスパラメータを表示し，ジャンプ先が転送前のアドレス範囲内にある場合，転送後のアドレスにリロケートします。転送前のソースプログラムはそのまま保存されます。データ文の混存で誤動作（データ内容を３バイト命令と間違えて変更する）を防止する為にデータ文の先頭にＦＤαβ（αβはデータ数）と書いておく事により以降のαβバイトの内容は転送後も保存されます。

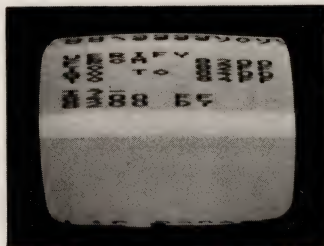
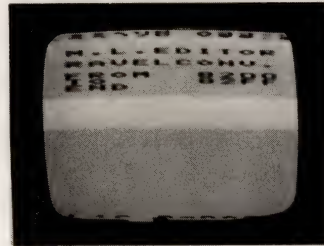
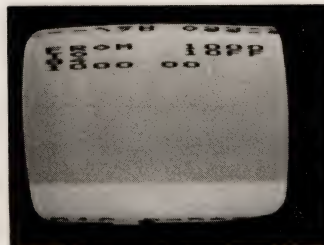
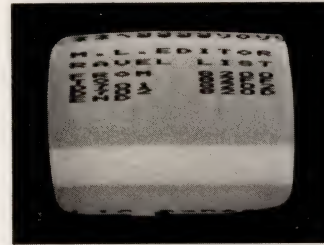
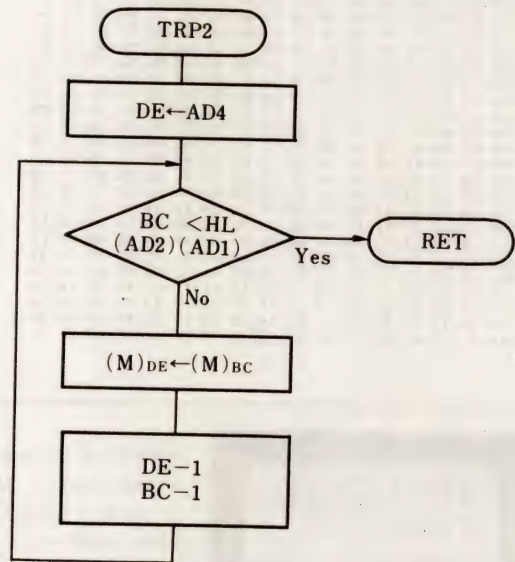
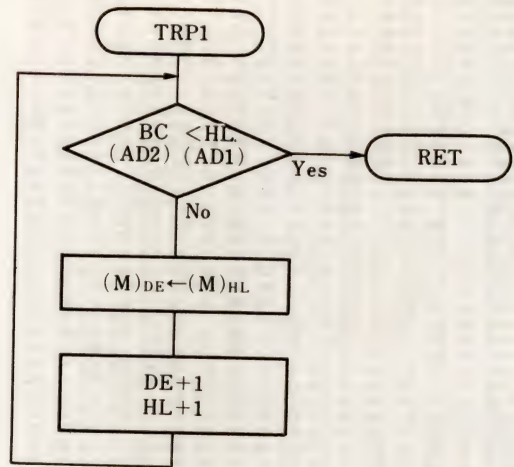
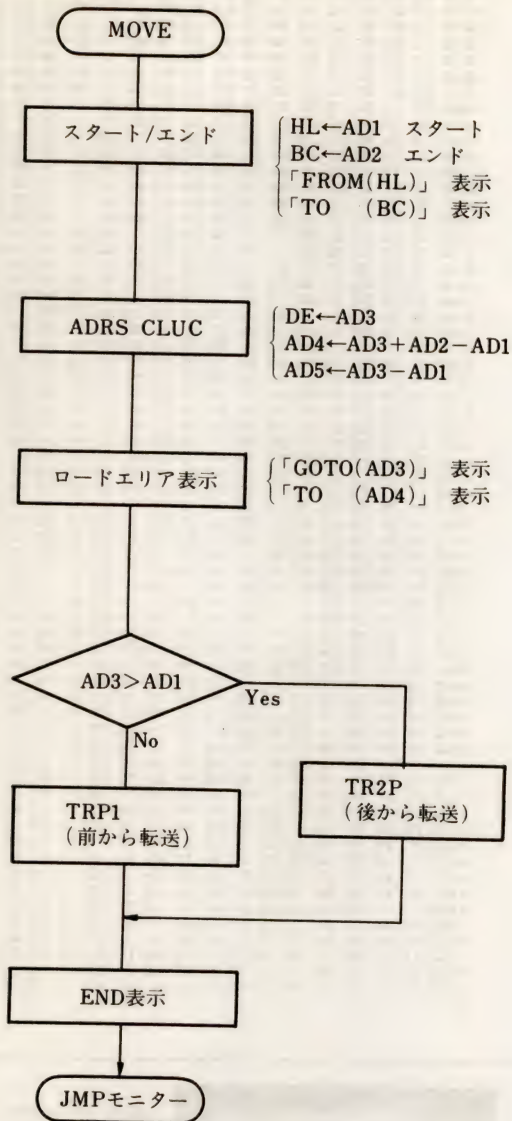
ラベルリスト，変換，削除はM. L. Editorのメインプログラムです。ソースプログラムを上記３回に分けて順次変更していく３パス方式です。デバッグ等の為に独立して実行出来る様にしました。本プログラムで編集する為には予め，サブルーチン等のジャンプ先の先頭にＤ７ｍｎ（ｍｎはラベルナンバー）を書込んでおき，ジャンプ先はＣ３ｍｎＤ７で指定します。パス１のラベルリストでＤ７ｍｎに対応する絶対アドレスがＴＶに表示されます。パス２のラベル変換で３バイト命令のジャンプ先ｍｎＤ７が絶対アドレスに変換されます。

さらにパス３のラベル削除でラベル名Ｄ７ｍｎが全て削除されて，以降のプログラムが前に詰められ，アドレスも２バイト分ずつリロケートされます。但し，ラベル削除の実行前のＤ７ｍｎをラベル名称として含んだままのソースプログラムのままでも，テスト的に実行できる様に，Ｄ７（ＲＳＴ２命令）後１バイト飛ばした次に進む様にＥＸ－８０のＲＳＴ２ジャンプテーブルの804Dにもプログラムが，ラベルリスト実行時に書込まれます。

相対アドレス方式を利用するにはジャンプ命令等のアドレスの代りにmnDB と書込むとｍｎバイト前の絶対アドレスに，mnDF と書込むとｍｎバイト後の絶対アドレスに，ラベル変換プログラム実行により置換されます。以上の変換はデータ文の宣言としてFDαβと書いた以降のαβバイト内で

M. L. Editor プログラムリスト

```
[#000 1A 13 A7 C8 FE FF CA 0F 08 CD DE 02 C3 00 08 CD
#010 60 03 C3 00 08 7C CD 11 03 7D CD 11 03 C9 7A CD
#020 11 03 7B CD 11 03 C9 78 CD 11 03 79 CD 11 03 C9
#030 CD 6C 03 11 7A 0F CD 00 08 C9 CD 6C 03 11 80 0F
#040 CD 00 08 C9 CD 6C 03 11 88 0F CD 00 08 C9 CD 6C
#050 03 11 B8 0F CD 00 08 11 94 0F CD 00 08 C9 CD 6C
#060 03 11 B8 0F CD 00 08 11 A0 0F CD 00 08 C9 CD 6C
#070 03 11 B8 0F CD 00 08 11 AC 0F CD 00 08 C9 FF FF
#080 2A A2 83 4D 44 2A A0 83 11 52 0F CD 00 08 CD 15
#090 08 11 59 0F CD 00 08 CD 27 08 2A A0 83 3A A2 83
#0A0 95 5F 3A A3 83 9C 57 2A A4 83 19 22 A6 83 2A A0
#0B0 83 3A A4 83 95 32 A8 83 3A A5 83 9C 32 A9 83 C9
#0C0 11 60 0F CD 00 08 2A A4 83 CD 15 08 E5 11 59 0F
#0D0 CD 00 08 2A A6 83 CD 15 08 2A A0 83 D1 C9 FF FF
#0E0 11 52 0F CD 00 08 2A A0 83 22 A4 83 CD 15 08 11
#0F0 59 0F CD 00 08 2A A2 83 22 A6 83 CD 15 08 C9 FF
#100 11 60 0F CD 00 08 2A A0 83 CD 15 08 11 59 0F CD
#110 00 08 2A A6 83 CD 15 08 00 11 74 0F CD 00 08 C9
#120 11 67 0F CD 00 08 C9 D5 11 6E 0F CD 00 08 D1 CD
#130 1E 08 CD 8F 03 1A CD 11 03 CD 60 03 CD 15 08 CD
#140 8F 03 7E CD 11 03 CD 69 00 FF FF FF FF FF FF FF
#150 79 95 78 9C D8 7E 7E 12 12 13 23 C3 50 09 EB 2A
#160 A6 83 EB 79 95 78 9C D8 0A 0A 12 12 0B 1B C3 63
#170 09 79 95 78 9C D8 1A BE C2 27 09 13 23 C3 71 09
#180 CD 3A 08 CD 80 08 CD C0 08 CD 71 09 CD 20 09 C9
#190 2A A4 83 EB 2A A6 83 7D 93 7C 9A D8 1A 13 FE FD
#1A0 CA EE 09 01 1A 13 21 E0 0F BE CA C5 09 23 0D C2
#1B0 A9 09 21 CD 0F BE CA C1 09 23 05 C2 B5 09 C3 94
#1C0 09 13 C3 94 09 1A 4F 13 1A 47 2A A0 83 13 79 95
#1D0 78 9C DA 94 09 2A A2 83 7D 91 7C 98 DA 94 09 2A
#1E0 A8 83 09 EB 2B 2B 73 23 72 23 EB C3 94 09 1A 13
#1F0 4F 06 00 EB 09 EB C3 94 09 FF FF FF FF FF FF E3
#200 23 E3 C9 3E C3 32 4D 80 21 FF 09 22 4E 80 21 00
#210 84 01 00 02 36 FF 23 0D C2 14 0A 05 C2 14 0A C9
#220 2A A0 83 EB 2A A2 83 7D 93 7C 9A D8 1A 13 FE FD
#230 CA 7B 0A 01 1A 12 21 E0 0F BE CA 76 0A 23 0D C2
#240 39 0A 21 CE 0F BE CA 72 0A 23 0D C2 45 0A FE D7
#250 C2 24 0A 1A 47 13 68 26 84 73 24 72 3E D7 CD 11
#260 03 78 CD 11 03 CD 8F 03 CD 8F 03 CD 1E 08 C3 24
#270 0A 00 13 C3 24 0A 13 13 C3 24 0A 1A 13 4F 06 00
#280 EB 09 EB C3 24 0A FF FF FF FF FF FF FF FF FF FF
#290 2A A0 83 EB 2A A2 83 7D 93 7C 9A D8 1A 13 FE FD
#2A0 CA C5 0A 01 1A 13 21 E0 0F BE CA D0 0A 23 0D C2
#2B0 A9 0A 21 CD 0F BE CA C1 0A 23 05 C2 B5 0A C3 94
#2C0 0A 13 C3 94 0A 1A 13 4F 06 00 EB 09 EB C3 94 0A
#2D0 13 1A FE D7 CA E5 0A FE DB CA 06 0B FE DF CA F6
#2E0 0A 13 C3 94 0A 1B 1A 6F 26 84 4E 24 46 79 12 13
#2F0 78 12 13 C3 94 0A 1B 1A 6F 26 00 1B 19 7D 12 7C
#300 23 12 23 C3 94 0A 1B 1A 6F 26 00 7B 95 6F 7A 9C
#310 67 2B 7D 12 7C 13 12 13 C3 94 0A FF FF FF FF FF
#320 06 00 2A A0 83 EB 2A A2 83 7D 93 7C 9A D8 2A A4
#330 83 78 AF C2 5F 0B 1A FE FD CA 6A 0B E5 01 1A 12
#340 21 E0 0F BE CA 70 0B 23 0D C2 43 0B 21 CE 0F BE
#350 CA 75 0B 23 05 C2 4F 0B 04 FE D7 CA 7A 0B E1 1A
#360 77 13 23 05 22 A4 83 C3 26 0B 13 1A 47 C3 5F 0B
#370 06 03 C3 5E 0B 06 02 C3 5E 0B 13 13 E1 22 A4 83
#380 2A A0 83 E5 EB 22 A0 83 CD 9A 08 CD 50 09 CD 90
#390 09 2A A0 83 EB E1 22 A0 83 06 00 C3 26 0B FF FF
#3A0 2A A0 83 2B 22 A4 83 C3 30 0C 2A A0 83 2B 22 A4
#3B0 83 C3 10 0C 2A A0 83 23 22 A4 83 C3 30 0C 2A A0
#3C0 83 23 22 A4 83 C3 10 0C FF FF FF FF FF FF FF FF
#3D0 21 00 84 22 A0 83 21 FF 9F 22 A2 83 21 00 A4 22
#3E0 A4 83 C3 00 0C 21 00 A4 22 A0 83 21 FF BF 22 A2
#3F0 83 21 00 84 22 A4 83 C3 00 0C FF FF FF FF FF FF]
#
```

```
#P.000,3FF
[#000 CD 90 0C CD B0 0C C3 69 00 FF FF FF FF FF FF FF
#010 CD 90 0C CD C0 0C C3 69 00 FF FF FF FF FF FF FF
#020 CD D0 0C CD E0 0C CD F0 0C C3 69 00 FF FF FF FF
#030 CD 90 0C C3 69 00 FF FF CD 80 09 C3 69 00 FF FF
#040 CD C0 0C C3 69 00 FF FF CD D0 0C C3 69 00 FF FF
#050 CD E0 0C C3 69 00 FF FF CD F0 0C C3 69 00 FF FF
#060 C3 A0 0B FF C3 AA 0B FF C3 B4 0B FF C3 BE 0B FF
#070 CD 64 0D C3 69 00 FF FF CD 6B 0D C3 69 00 FF FF
#080 C3 D0 0B FF FF FF FF FF C3 E5 0B FF FF FF FF FF
#090 CD 30 08 CD 80 08 CD C0 08 7D 93 7C 9A D2 A7 0C
#0A0 CD 5E 09 CD 19 09 C9 CD 50 09 C3 A3 0C FF FF FF
#0B0 CD 30 08 CD 80 08 CD C0 08 CD 71 09 CD 20 09 C9
#0C0 CD 44 08 CD 80 08 CD C0 08 CD 90 09 CD 19 09 C9
#0D0 CD 4E 08 CD 03 0A CD E0 08 CD 20 0A CD 19 09 C9
#0E0 CD 5E 08 CD E0 08 CD 90 0A CD 19 09 C9 FF FF FF
#0F0 CD 6E 08 CD E0 08 CD 20 0B CD 00 09 C9 FF FF FF
#100 CD 6C 03 2A B2 83 EB 2A B0 83 3A B4 83 47 D5 11
#110 52 0F CD 00 08 CD 15 08 11 59 0F CD 00 08 D1 CD
#120 1E 08 78 CD 11 03 C9 7B 95 7A 9C DA 33 0D 70 23
#130 C3 27 0D 2A B2 83 EB 2A B0 83 3A B4 83 47 2B 23
#140 7E B8 C2 53 0D 7B BD C2 3F 0D 7A BC C2 3F 0D CD
#150 20 09 C9 11 7E 0F CD 00 08 CD 15 08 CD 8F 03 7E
#160 CD 11 03 C9 CD 00 0D CD 27 0D C9 CD 00 0D CD 33
#170 0D C9 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#1F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#200 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#210 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#220 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#230 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#240 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#250 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#260 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#270 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#280 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#290 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#2F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#300 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#310 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#320 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#330 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#340 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
#350 FD 90 06 12 0F 0D 20 20 00 14 0F 20 20 20 20 00
#360 07 0F 20 14 0F 20 00 FF 07 0F 0F 04 FF 00 FF 74
#370 57 70 FF 00 FF 05 0E 04 FF 00 0D 0F 16 05 FF 00
#380 16 05 12 09 06 19 FF 00 01 04 12 13 03 08 01 0E
#390 07 05 FF 00 12 01 16 05 0C 20 0C 09 13 14 FF 00
#3A0 12 01 16 05 0C 03 0F 0E 16 2E FF 00 12 01 16 05
#3B0 0C 05 12 01 13 05 FF 00 0D 2E 0C 2E 05 04 09 14
#3C0 0F 12 FF 00 FF FF FF FF FF FF FF FF FF D7 06 0E
#3D0 16 1E 26 2E 36 3E C6 CE D3 D6 DB DE E6 EE F6 FE
#3E0 01 11 21 22 2A 31 32 3A C2 C3 C4 CA CC CD D2 D4
#3F0 DA DC E2 E4 EA EC F2 F4 FA FC FF FF FF FF FF FF]
#
```

0C00 Move & Verifyで(8200～84FF)を(8400～84FF)に転送後のVerifyで8400がエラー(FF)を発見

0C48 ラベル変換実行後

0C70 Write & Verifyで(1000～10FF)に84を書込後のVerifyで1000がエラー(00)を発見

0C50 ラベルList実行例
ラベルD701の絶対アドレスは8202
ラベルD702の絶対アドレスは820D

は何も実行されません。

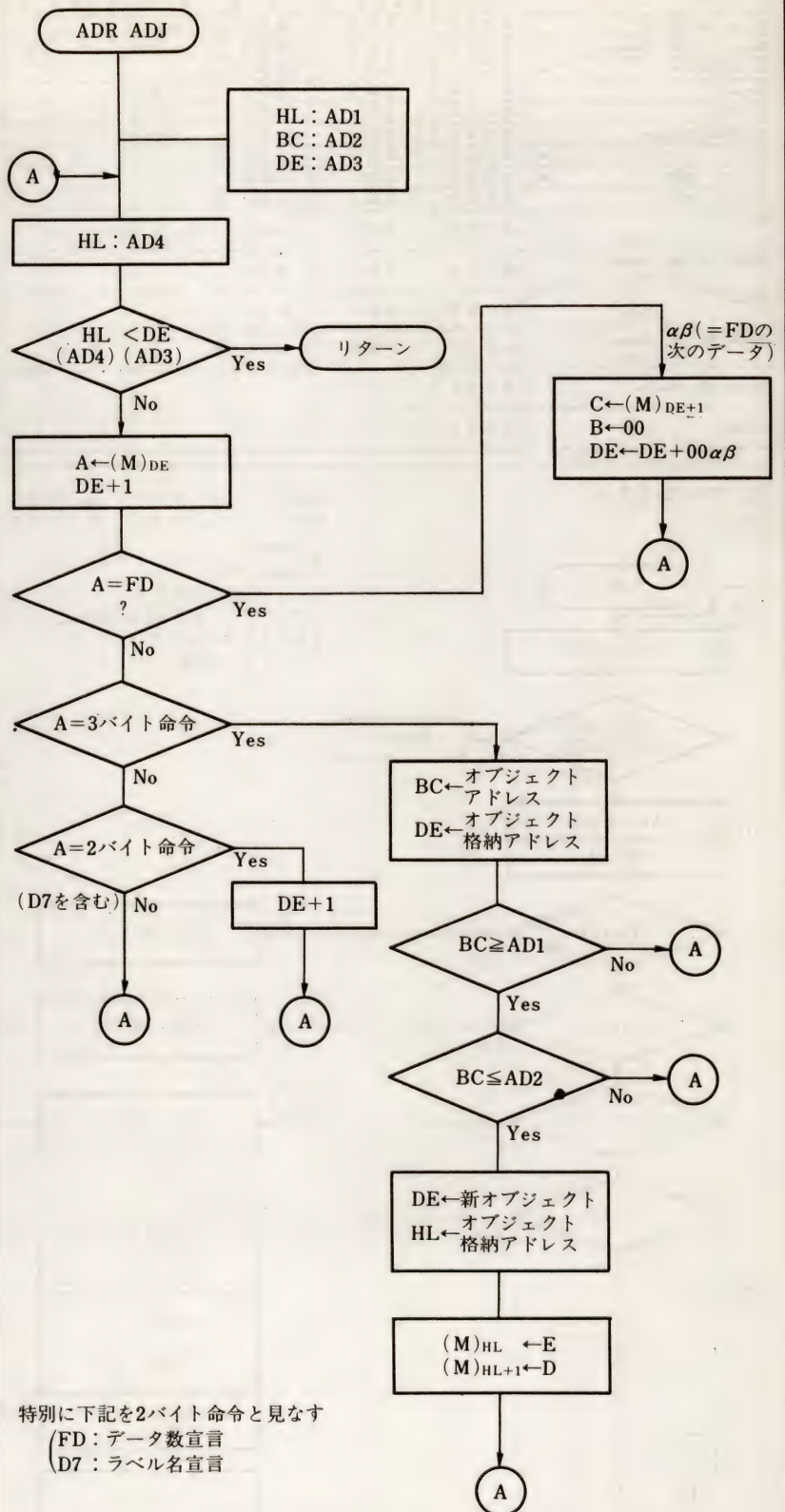
よく使う1バイト毎の前後の移動及びアドレスのリロケートが簡単に呼び出せる様に各々のスタートプログラムを用意しました。指定した範囲内に指定データを書込み，Verifyするプログラムも追加してあります。アドレスの指定のミスでソースプログラムがこわされる事を少しでも防止する為に，スタートアドレスがエンドアドレスよりも大きい場合はプログラムは実行されません。本プログラムで編集可能なアドレス範囲は0001～FFFEです。本プログラムリスト中には他のプログラムが一緒に載っています（4581～4749）が本説明とは無関係です。

使　用　方　法

第4－1表にアドレス範囲を予めセットするべきアドレスとスタートアドレスを示します。アドレスは全て下位を先に上位を後からセットします。セットされたパラメータは保存されます。転送先

のエンドアドレス等は計算されＴＶにFROM××××ＴＯ××××と表示されます。

４Ｋバイト程度のプログラムの転送も数秒で実行されます。ラベル変換は，含まれるラベルの数により実行時間が変ります。

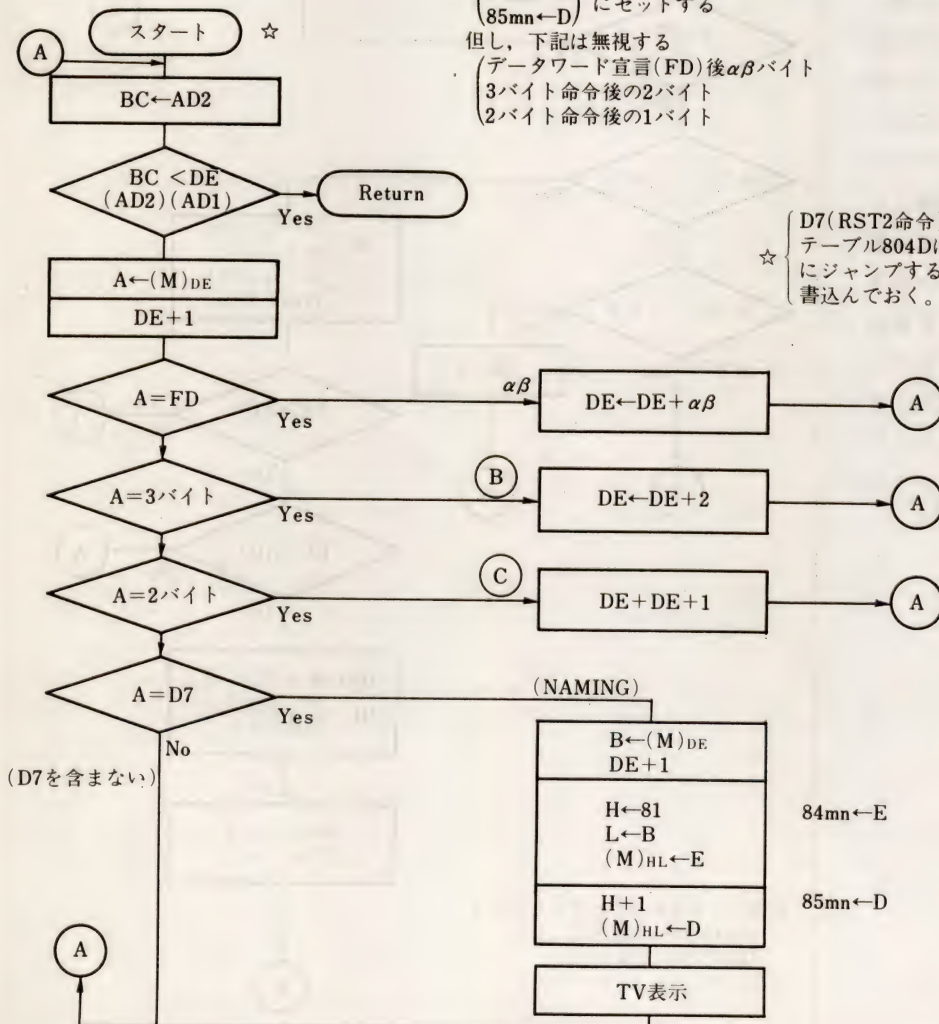
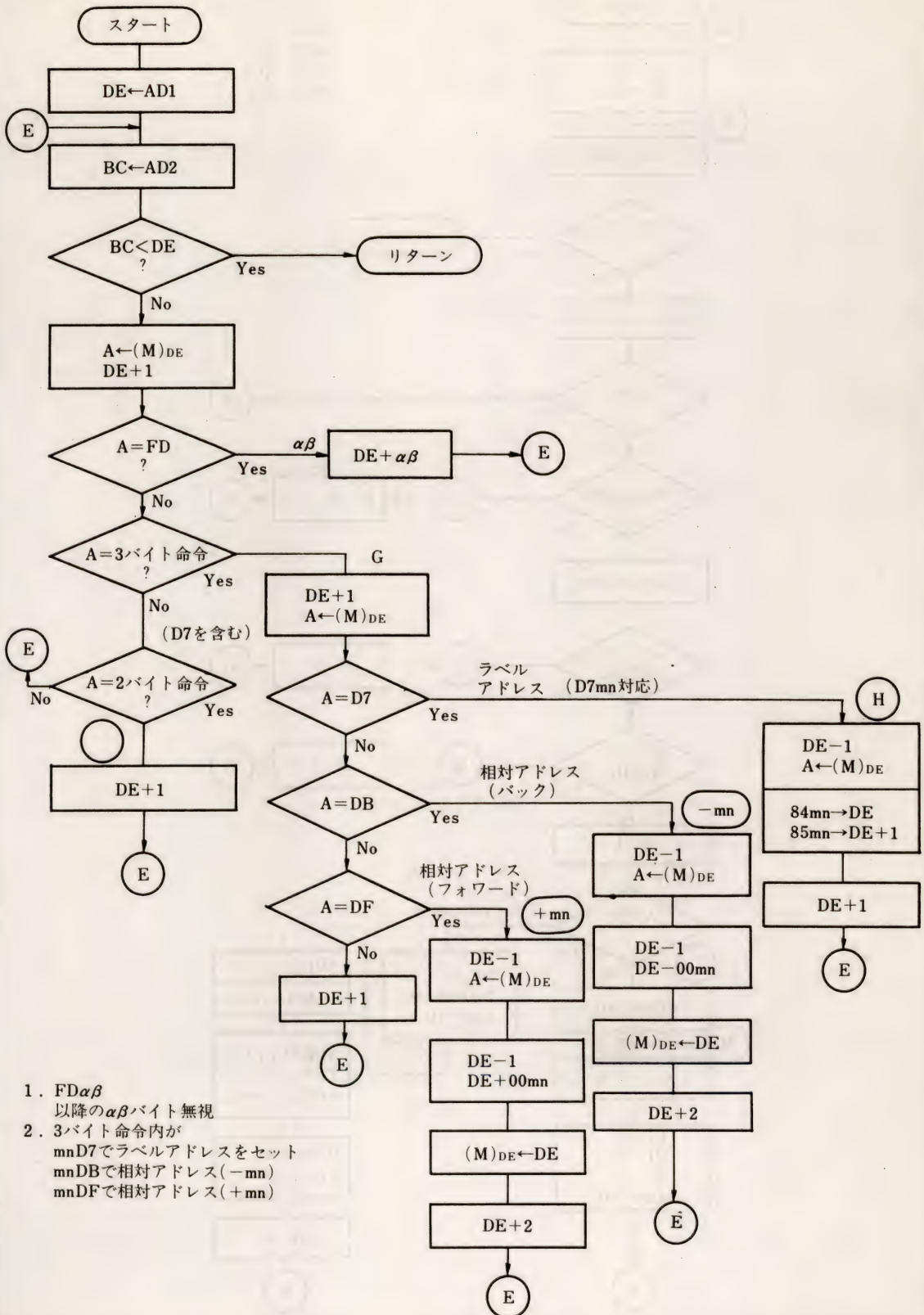
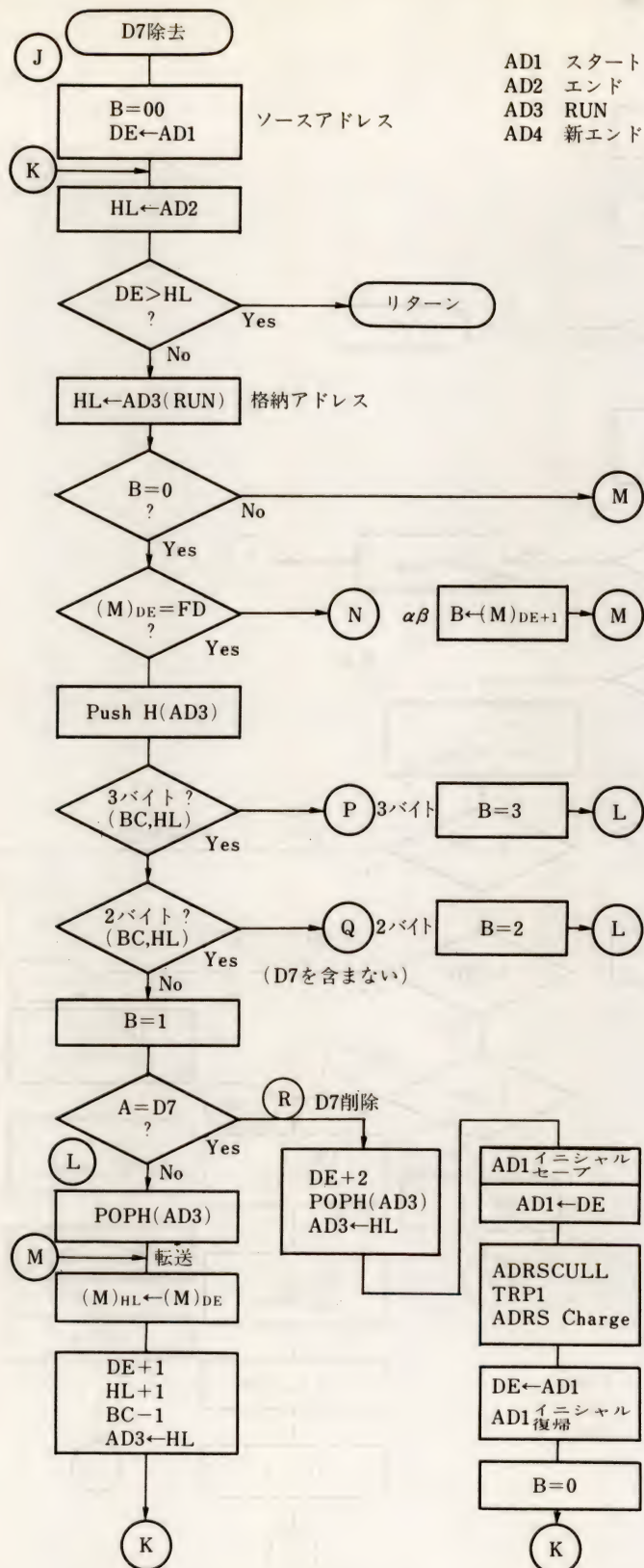
機械語でプログラムを作るには本格的なアセンブラが当然便利ですが，フルキー・ボード，プリンタ等が必要です。今回紹介のM. L. Editorは16進のキーボードしか持っていないアマチュアの同好の方々に少しでもお役に立てば幸と考えて作りました。なお，本プログラムは，Ｐ－ＲＯＭ化して東芝マイコンセブン林氏にお預り頂いています。ＴＭＭ３２２Ｃを２個送って頂けば私の自宅でも書込みをして差し上げますので、どうぞテストして見て下さい。

《第4－1表》M. L. Editor　使用方法

実行内容	スタートアドレス	パラメータ用RAM 83A0　1	2　3	4　5	83B0　1	2　3	4　5	備考
Move & Verify	0C00	AD1	AD1	AD3				AD1 ソーススタート AD2 ソースエンド AD3 転送先 スタート ★印の ワーキング ラム 8400 ～85FF
Move & AD change	0C10	AD1	AD2	AD3				
M. L. Editor ★	0C20	AD1	AD2					
Move	0C30	AD1	AD2	AD3				
Verify	0C38	AD1	AD2	AD3				
ADRES change	0C40	AD1	AD2	AD3				
ラベル　リスト ★	0C48	AD1	AD2					
ラベル　変換 ★	0C50	AD1	AD2					
ラベル　削除	0C58	AD1	AD2					
Move（1つ前）	0C60	AD1	AD2					
Move & AD change（1つ前）	0C64	AD1	AD2					
Move（1つ後）	0C68	AD1	AD2					
Move & AD change（1つ後）	0C6C	AD1	AD2					
Write & Verify	0C70				スタートアドレス	エンドアドレス	データ	
Verify	0C78				スタートアドレス	エンドアドレス	データ	

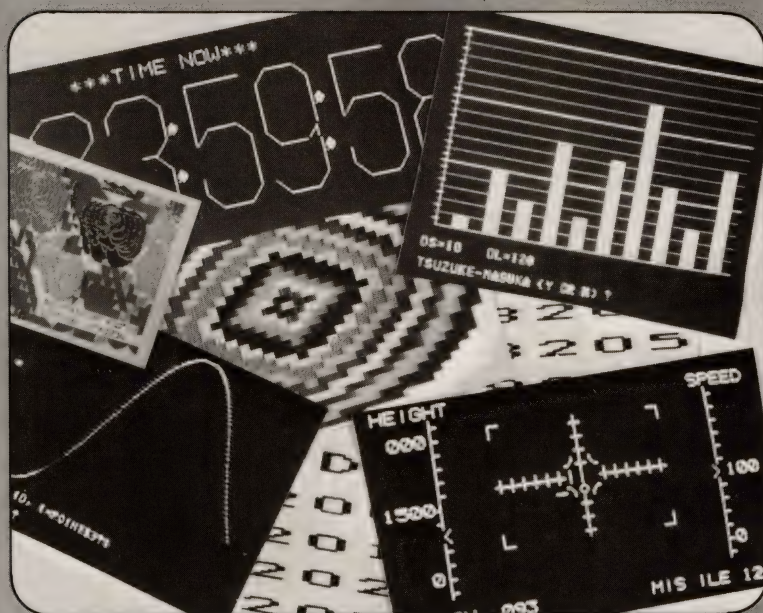
3．ラベルリスト

4．ラベル→アドレス変換
スタート
DE←AD1
E
BC←AD2
BC<DE
?
Yes
リターン
No
A←(M)DE
DE+1
A=FD
?
Yes
αβ
DE+αβ
E
No
A=3バイト命令
?
Yes
G
No
DE+1
A←(M)DE
(D7を含む)
E
A=2バイト命令
?
No
Yes
A=D7
Yes
ラベル
アドレス　(D7mn対応)
H
DE+1
No
DE−1
A←(M)DE
A=DB
Yes
相対アドレス
(バック)
−mn
84mn→DE
85mn→DE+1
E
No
DE−1
A←(M)DE
相対アドレス
(フォワード)
A=DF
Yes
+mn
DE+1
No
DE−1
A←(M)DE
DE−1
DE−00mn
E
DE+1
E
(M)DE←DE
DE−1
DE+00mn
DE+2
E
(M)DE←DE
E
DE+2
E
1．FDαβ
以降のαβバイト無視
2．3バイト命令内が
mnD7でラベルアドレスをセット
mnDBで相対アドレス(−mn)
mnDFで相対アドレス(+mn)

5．ラベル削除

D7除去
J
B=00
DE←AD1
ソースアドレス
AD1 スタート
AD2 エンド
AD3 RUN
AD4 新エンド
K
HL←AD2
DE>HL ?
Yes
リターン
No
HL←AD3(RUN)
格納アドレス
B=0 ?
No
M
Yes
(M)DE=FD ?
N
αβ
B←(M)DE+1
M
Yes
Push H(AD3)
3バイト ? (BC,HL)
P
3バイト
B=3
L
Yes
2バイト ? (BC,HL)
Q
2バイト
B=2
L
Yes
No
(D7を含まない)
B=1
A=D7 ?
R
D7削除
Yes
L
No
DE+2
POPH(AD3)
AD3←HL
AD1 イニシャル セーブ
AD1←DE
POPH(AD3)
M
転送
(M)HL←(M)DE
ADRSCULL
TRP1
ADRS Charge
DE+1
HL+1
BC−1
AD3←HL
DE←AD1
AD1 イニシャル 復帰
B=0
K
K

第3章

各種マイコン
応用プログラム集

5

効果音プログラム付き

インベータ・ゲーム

TK-80BSL II

野口武志

このゲームは最近ゲームセンター等で人気のある「スペース・インベーダー」のTK-80BS版です。

ゲーム・センター等のこのゲームは縦長の画面を使い，ビーム砲は左右に動きますが，BSは32(横)×16(縦)ですので，ビーム砲は右端で上下するようにしました。また画面の左端に得点とビーム砲の残りの数と最高得点を常に表示しています。

ゲームのルール

○インベーダー(2AH)は20点，時々現われるUFO(24H)は100点です。ゲームはビーム砲が3回攻撃を受けるか，インベーダーがこちらの右端の領域にはいると終了となります。ただし得点が1500点を超えるとビーム砲が1つ追加されます。

○UFOは時々画面の左端を上から下へ（または下から上へ）現われ，移動しますので，インベーダーの間をぬって攻撃してください。

○インベーダーは上下に動きます。上または下につかえると1段右へ移動します。また画面のインベーダーをすべて撃破（消してしまう）すると，しばらくしてさらに右によった位置にインベーダが現われ，ゲームは続きます。

インベーダーは横6×縦8の48個ですがその左端の位置がカーソルのXで10，次は14さらに16，最後は18とだんだん右へ移動します。

○ビームはインベーダーが一つ，こちらは二つですので，それらが画面より消えるまで，次のビームを発射することはできません。

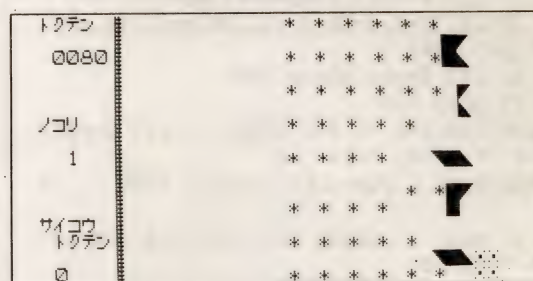
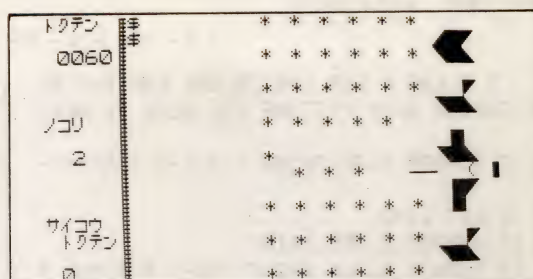
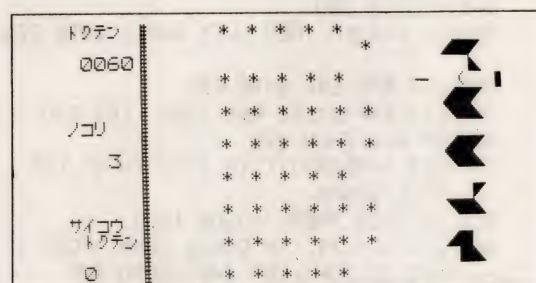
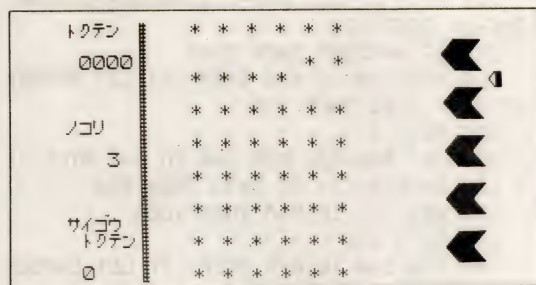
そして，インベーダーのビームと，こちらのビームがぶつかれば，こちらのビーム砲は必ず負けますので，ビーム砲を動かし，逃げてください。

○ビーム砲の近くにある防御ブロックはビームが

```
******************
* スペース インベーダー *
******************




スタート ボタン(S) ヲ オシテクダサイ !_
```

《写真5—1》ゲーム開始

```
*************
* GAME OVER *
*************


アナタ ノ トクテン ハ サイコウ デス !!

サイコウ トクテン ・・・ 80

スタート ボタン(S) ヲ オシテクダサイ !_
```

《写真5—2》ゲーム終了

当るとかけていきます。

○音はビーム発射時，インベーダー撃破時，UFO移動時，UFO撃破時，ビーム砲撃破時，ビーム砲追加時，こちらの領域にインベーダーがはいりゲームが終了する時に発生します。

ゲームのやり方

プログラムを入力し，RUNさせてください。「タイトル」と「スタートボタン (S) ヲオシテクダサイ！」が表示されますのでBSのフルキー・ボードの「S」を押して下さい。すると，ゲームはスタートします。

ゲーム中は数字キー1，2，3，0のみで進行します。

1：ビーム砲を下へ移動

2：ビーム砲移動停止，ビーム発射停止

3：ビーム砲を上へ移動

0：ビーム発射

プログラムの説明

5～8：タイトル表示，最高得点のクリア

10～99：初期値設定，ゲーム開始待ち

100～225：メインプログラム

100～118＝インベーダーが下につかえた時，右へ移動

120～155＝インベーダーの上への移動

200～218＝インベーダーが上につかえた時，右へ移動

220～235＝インベーダーの下への移動

500～666：キー入力によりビーム砲の移動，ビームの発射，そしてビームの移動，UFOの移動など

1000〜1020：画面のインベーダーがすべて消えた時，次のインベーダーの位置の決定

1500〜1550：ビーム砲の撃破

1600〜1640：UFOの撃破

2000〜2100：ゲーム終了時の画面表示，そして行番号10へとぶ

3000〜3100：ゲームスタート時及びインベーダーがすべて消えた時，次の画面を表示

スペースインベーダーゲームプログラムリスト

```
   5 CLEAR : FOR X=8 TO 25: FOR Y=2 TO
 4: CURSOR X,Y: PICTURE 2A: NEXT Y: NEXT
 X
   7 CURSOR 9,3: PRINT " ｽﾍﾟｰｽ ｲﾝﾍﾞｰﾀﾞｰ
";
   8 LET L1=0
  10 CURSOR 3,15: PRINT
  12 CURSOR 3,16: PRINT "ｽﾀｰﾄ ﾎﾞﾀﾝ(S) ｦ
 ｵｼﾃｸﾀﾞｻｲ !";
  14 LET H=PEEK(7DFCH): IF H<>83 THEN 1
4
  16 RANDOMIZE
  20 LET W4=32321,W5=W4+1,W6=W5+1,N=0
  30 LET F=0,I=50,G=16,J=0,K=0,L=0,M=0,
O=3,M=0,O1=1
  50 LET Z=10: GOSUB 3000
  99 GOTO 120
 100 FOR C=B TO B-E STEP -1: LET D=PEEK
(C): IF D<>42 THEN 110
 102 POKE C,20H: LET C1=C+2: POKE C1,2A
H
 110 NEXT C: GOSUB 500: LET B=B-64: IF
B>A THEN 100
 115 LET B=B1+2,B1=B,A=A1+2,A1=A
 116 FOR C=A TO B STEP 32: LET D=PEEK(C
): IF D=42 THEN 118
 117 NEXT C: LET A=A-2,A1=A,B=B-2,B1=B,
E=E-2: GOTO 120
 118 IF A=32287 THEN 2000
 120 FOR C=A TO A-E STEP -1: LET D=PEEK
(C): IF D=42 THEN 200
 125 NEXT C
 130 LET A=A+32: FOR C=A TO A-E STEP -1
: LET D=PEEK(C): IF D=42 THEN 140
 132 NEXT C: IF A=B THEN 1000
 134 GOTO 130
 140 FOR C=A TO A-E STEP -1: LET D=PEEK
(C): IF D<>42 THEN 150
 142 POKE C,20H: LET C1=C-32: POKE C1,2
AH
 150 GOSUB 500: NEXT C: IF A=B THEN  LE
T A=A1: GOTO 120
 152 IF A+32=B THEN  LET A=A1: GOTO 120

 155 LET A=A+64: GOTO 140
 200 FOR C=A TO A-E STEP -1: LET D=PEEK
(C): IF D<>42 THEN 210
 202 POKE C,20H: LET C1=C+2: POKE C1,2A
H
 210 NEXT C: GOSUB 500: LET A=A+64: IF
A<B THEN 200
 215 LET A=A1+2,A1=A,B=B1+2,B1=B
 216 FOR C=A TO B STEP 32: LET D=PEEK(C
): IF D=42 THEN 218
 217 NEXT C: LET A=A-2,A1=A,B=B-2,B1=B,
E=E-2: GOTO 220
 218 IF A=32287 THEN 2000
 220 FOR C=B TO B-E STEP -1: LET D=PEEK
(C): IF D=42 THEN 100
 225 NEXT C
 230 LET B=B-32: FOR C=B TO B-E STEP -1
: LET D=PEEK(C): IF D=42 THEN 240
 232 NEXT C: IF B=A THEN 1000
 234 GOTO 230
 240 FOR C=B TO B-E STEP -1: LET D=PEEK
(C): IF D<>42 THEN 250
 242 POKE C,20H: LET C1=C+32: POKE C1,2
AH
 250 GOSUB 500: NEXT C: IF B=A THEN  LE
T B=B1: GOTO 220
 252 IF B-32=A THEN  LET B=B1: GOTO 220

 255 LET B=B-64: GOTO 240
 500 LET F=F+1: IF F=6 THEN  LET F=0
 510 IF J=0 THEN 520
 512 LET D=PEEK(J): IF D=42 THEN  LET L
=L+2: CALL 8200H
 513 IF D=36 THEN  GOSUB 1600
 514 LET J2=J+1: IF D<>32 THEN  POKE J,
20H: POKE J2,20H: LET J=0: GOTO 520
 516 POKE J,28H: POKE J2,20H: IF J=J1 T
HEN  POKE J,20H: LET J=0: GOTO 520
 518 LET J=J-1
 520 IF K=0 THEN 530
 522 LET D=PEEK(K): IF D=42 THEN  LET L
=L+2: CALL 8200H
 523 IF D=36 THEN  GOSUB 1600
 524 LET K2=K+1: IF D<>32 THEN  POKE K,
20H: POKE K2,20H: LET K=0: GOTO 530
 526 POKE K,28H: POKE K2,20H: IF K=K1 T
HEN  POKE K,20H: LET K=0: GOTO 530
 528 LET K=K-1
 530 IF F<>2 THEN 540
 532 IF M<>0 THEN 540
 534 LET M1=32254+G*32: FOR M=M1 TO M1-
20 STEP -1
 536 LET D=PEEK(M): IF D=42 THEN  LET M
=M+1: GOTO 540
 538 NEXT M: LET M=0
 540 IF M=0 THEN 550
 542 LET D=PEEK(M): IF D=192 THEN 1500
 543 IF D=130 THEN 1500
 544 LET M2=M-1: IF D<>32 THEN  POKE M,
20H: LET M=0: GOTO 548
 545 POKE M,9AH: IF M2=M1 THEN  POKE M,
20H: LET M=0: GOTO 548
 547 LET M=M+1
 548 LET D=PEEK(M2): IF D<>42 THEN  POK
E M2,20H
```

```
 550 IF F<>0 THEN 560
 551 IF N<>0 THEN 555
 552 IF RND<.95 THEN 555
 553 IF RND<.5 THEN  LET N=32262,N1=327
42,N2=32: GOTO 557
 554 LET N=32742,N1=32262,N2=-32: GOTO
557
 555 IF N=0 THEN 560
 556 POKE N,20H: LET N=N+N2
 557 IF N=N1 THEN  POKE N,20H: LET N=0:
 GOTO 560
 558 POKE N,24H: LET N3=N+N2: POKE N3,2
4H
 559 CALL 8200H: CALL 8206H: CALL 8200H

 560 LET W1=INT(L/100),W2=INT((L-W1*100
)/10),W3=L-W1*100-W2*10
 562 LET W1=W1+48,W2=W2+48,W3=W3+48
 565 POKE W4,W1: POKE W5,W2: POKE W6,W3

 570 IF O1=0 THEN 600
 575 IF L<150 THEN 600
 580 FOR T=1 TO 10: CALL 8206H: CALL 82
00H: NEXT T: LET O1=0,O=O+1
 585 CURSOR 2,9: PRINT O;
 600 LET H=PEEK(7DFCH): IF H<48 THEN  L
ET H=I
 610 IF H>51 THEN  LET H=I
 620 LET I=H: ON H-47 GOTO 660,630,640,
650
 630 IF G<16 THEN  CURSOR 31,G: PICTURE
 20,20: LET G=G+1
 640 CURSOR 31,G: PICTURE C0,82: RETURN

 650 IF G>1 THEN  CURSOR 31,G: PICTURE
20,20: LET G=G-1
 652 GOTO 640
 660 IF F<>5 THEN  RETURN
 662 IF J=0 THEN  LET J=32253+G*32,J1=J
-23: CALL 8206H: RETURN
 664 IF K=0 THEN  LET K=32253+G*32,K1=K
-23: CALL 8206H: RETURN
 666 RETURN
 1000 IF Z=10 THEN  LET Z=14: GOTO 1010
 1002 IF Z=14 THEN  LET Z=16: GOTO 1010
 1003 IF Z=16 THEN  LET Z=18
 1010 LET G=16: CLEAR : GOSUB 3000
 1020 GOTO 120
 1500 LET G1=G-1,G2=G+1: CURSOR 30,G: PI
CTURE 20,FA,FA
 1505 CURSOR 31,G1: PICTURE FA,FA: CURSO
R 31,G2: PICTURE FA,FA
 1520 FOR T=1 TO 10: CALL 8200H: CALL 82
06H: NEXT T
 1525 IF O=1 THEN 2005
 1530 LET O=O-1: CURSOR 2,9: PRINT O;
 1540 CURSOR 30,G: PICTURE 20,20,20: LET
 G=16
 1545 CURSOR 31,G1: PICTURE 20,20: CURSO
R 31,G2: PICTURE 20,20
 1550 LET M=0: GOTO 550
 1600 LET L=L+10,N4=N+1,N5=N3+1
 1610 POKE N,FAH: POKE N3,FAH: POKE N4,F
AH: POKE N5,FAH
 1620 FOR T=1 TO 5: CALL 8200H: CALL 820
6H: NEXT T
 1630 POKE N,20H: POKE N3,20H: POKE N4,2
0H: POKE N5,20H
 1640 LET N=0: RETURN
 2000 FOR X=31 TO 32: FOR Y=1 TO 16: CUR
SOR X,Y: PICTURE FA: NEXT Y: NEXT X
 2002 FOR T=1 TO 10: CALL 8200H: CALL 82
06H: NEXT T
 2005 FOR T=1 TO 300: NEXT T: CLEAR
 2010 FOR X=10 TO 22: FOR Y=2 TO 4: CURS
OR X,Y: PICTURE 2A: NEXT Y: NEXT X
 2020 CURSOR 11,3: PRINT " GAME OVER";
 2025 CURSOR 4,8: PRINT
 2030 CURSOR 4,9: PRINT "ｱﾅﾀ ﾉ ﾄｸﾃﾝ ･･･"
;L*10
 2040 IF L>L1 THEN  LET L1=L: CURSOR 15,
9: PRINT "ﾊ ｻｲｺｳ ﾃﾞｽ !!"
 2050 CURSOR 5,12: PRINT "ｻｲｺｳ ﾄｸﾃﾝ ･･･"
;L1*10
 2100 GOTO 10
 3000 CLEAR : PRINT "ﾄｸﾃﾝ": CURSOR 1,7:
PRINT "ﾉｺﾘ": CURSOR 1,13: PRINT "ｻｲｺｳ":
PRINT " ﾄｸﾃﾝ"
 3002 CURSOR 5,3: PRINT "0"
 3003 CURSOR 3,9: PRINT O;: CURSOR 1,16:
 PRINT L1*10;
 3010 LET A=32265+Z,A1=A,B=A+480,B1=B,E=
10
 3020 FOR Y=1 TO 16: CURSOR 6,Y: PICTURE
 CF: NEXT Y
 3030 FOR Y=2 TO 14 STEP 3: CURSOR 28,Y:
 PICTURE B2,80,B1: LET Y1=Y+1
 3040 CURSOR 28,Y1: PICTURE B5,80,B4: NE
XT Y
 3050 FOR X=Z TO Z+10 STEP 2: FOR Y=2 TO
 16 STEP 2: CURSOR X,Y
 3060 PICTURE 2A: NEXT Y: NEXT X
 3100 RETURN
```

```
 8200  00 82 35 82 09 82 01 05 05 F5 C5
D5 E5 2E FF 61
  8210  48 3E 02 D3 FA 2B 7C A7 CA 31 82
0D C2 15 82 48
  8220  3E 00 D3 FA 2B 7C A7 CA 31 82 0D
C2 24 82 C3 10
  8230  82 E1 D1 C1 F1 C9
```

6

カラーグラフィックによる

花火シミュレーション

LKIT-16

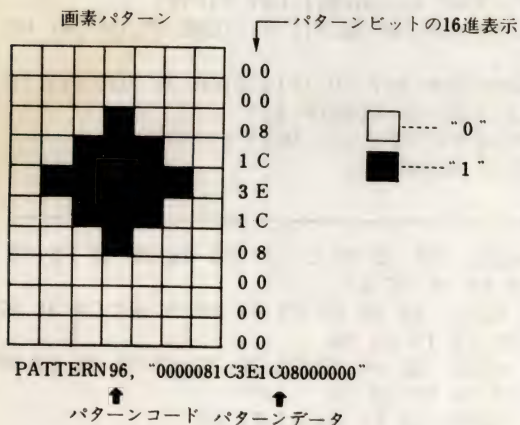
原田雅英

花火のおもしろさは，何といっても色の美しさと豪快な打上げ音にあります。そこで，色，音のすべてを動員することにします。

LKIT-16 BASIC

色も音も出せるBASIC，一体どこが普通のBASICと違っているのか，まず最初にそこら辺から紹介しましょう。

PATTERN文

花火の一片のように画面に表示する一つの画素の形を決定するための文で，画素コード（10進）の指定，画素コード（16進）の記述を行います。画素は**第6－1図**のように縦10ドット，横7ドット

《第6－1図》PATTERN文と画素パターン

からなりますが，ここのビット配置をこのPATTERN文で記述しておけば，テレビインターフェース内のパターンメモリにビットデータが記憶され，あとでパターンコードを出力すれば，その形の画素が表示されるのです。

CHR$関数

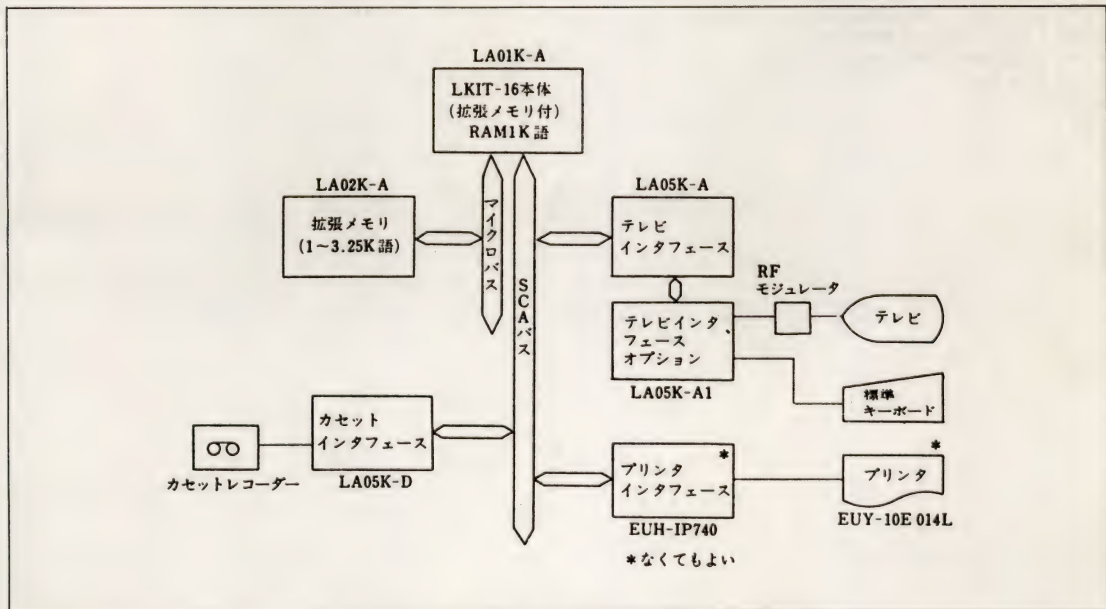
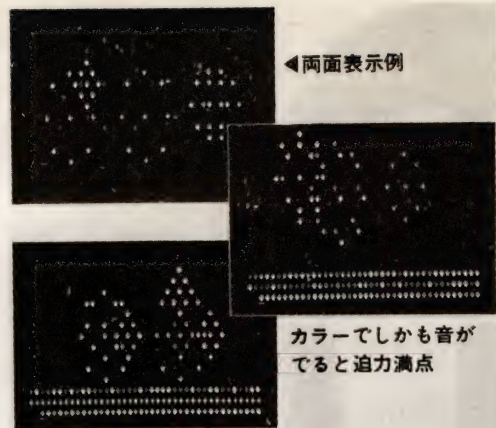
この関数そのものは，他のBASICにも備えられている一般的な機能ですが，この関数を使って，テレビインターフェースの吸うあらゆるコードが出力できる点が一味違うところです。

表示する画素の色指定，効果音の出力はこのCHR$関数を使って色コード，音出力コードを出力することにより行われます。カナ文字の表示も同様な方法で行います。Lkit-16BASIC全体の仕様はANSI規格の最小BASIC案に準拠した，互換性のある仕様になっています。**第6－1表**にその全体を示しておきます。

このBASICのもう一つの特色は，1行多文(マルチ文）機能の他に，各種キー・ワードの省略形記述が許されていることで，長大なプログラムをコンパクトにまとめる，つまりメモリの使用効率を上げることができます。

Lkit-16BASICを使用するには，**第6－2図**のような機器構成が必要です。

項　目	内　容
数値データ	0および絶対値が9.99999×10^{-39}~$1,70141 \times 10^{38}$の数
文字列データ	最大18文字までの文字列
数値定数	6 桁以内の指数部なし数，6 桁以内の仮数部+指数部
文字定数	"　"で囲まれた18文字以内の文字列
変数，配列名	英字1字または英字1字に数字1字をつける
文字変数名	英字1字に"$をつける
算術演算子	*　/　+　−
比較演算子	>　=　<　>=　<=　<>
数値関数	SIN, COS, ATN, SQR, EXP, LOG, INT, ABS, RND
出力文関係	CHR$($n_1$, …, n_j)　TAB(n)　CSR(i, j)　CSR·カーソル
定義関数	FNa(X)　a：英字1字
基本文 (ANSI規格案に準拠)	LET, PRINT, GOTO, INPUT, IF THEN, ON GOTO FOR, NEXT, GOSUB, RETURN, END, STOP, DATA READ, RESTOR, DEF, ANAMIZE, DIM(2次元 REM
拡張文	ERASE　テレビ画面を消去する PATTERN　画素コードに画素データを割付ける TV　出力機器にテレビを指定する PR　出力機器にプリンタを指定する CALL　機械語プログラムへサブルーチン分岐する GMODE　グラフィックモードを指定する 注1 CMODE　キャラクタモードを指定する 注2
指令	RUN　プログラムの実行開始 NEW　プログラムの消去 SAVE　プログラムをカセットへ保存 LOAD　カセットからプログラムを格納 LIST　プログラムリストの作成 MAP　メモリサイズなどの指定
	注1：画面上の表示位置がCSR関係により指定できるモード 注2：スクロールが行われるモード

《第6−1表》LKIT-16BASICの機能概要

《第6−2図》LKIT-16BASICに必要な機器構成

《第6-2表》花火シミュレーションプログラム

```
 10 PATTERN 96,"0000081C3E1C08000000"
 15 Y=INT(RND*3)+3
 17 FOR Q=0 TO Y
 20 GMODE:PRINT CHR$(144):ERASE
 25 K=INT(RND*6)+4
 30 FOR N=1 TO K
 40 X=INT(RND*32)+4:Y=INT(RND*5)+4
 50 H=INT(RND*3)+1:C=INT(RND*7)*2+146
 60 GOSUB 500
 65 P=INT(RND*100):GOSUB 250
 70 NEXT N
 71 P=50
 72 GOBUS 250
 75 NEXT Q
 87 GOTO 350
250 FOR L=1 TO P
260 NEXT L
270 RETURN
350 FOR X=0 TO 39
360 PRINT CSR(X,13);CHR$(158,96)
362 PRINT CSR(X,14);CHR$(156,96)
364 PRINT CSR(X,15);CHR$(154,96)
366 NEXT X
370 C=154
380 FOR P=0 TO 2
385 Y=13:GOBUS 400
387 Y=14:GOBUS 400
390 Y=15:GOBUS 400
392 NEXT P
395 GOTO 15
400 IF C<>152 THEN 410:C=160
410 C=C-2
420 FOR K=0 TO 40
430 X=INT(RND*40)
440 PRINT CSR(X,Y);CHR$(C,96)
450 NEXT K
460 RETURN
500 PRINT CHR$(2);CSR(X,Y):CHR$(C,96)
510 M=H
520 PRINT CSR(X,Y-1-M):CHR$(96)
530 PRINT CSR(X+M,Y-M);CHR$(96)
540 PRINT CSR(X+1+M,Y);CHR$(96)
550 PRINT CSR(X+M,Y+M);CHR$(96)
560 PRINT CSR(X,Y+M+1);CHR$(96)
570 PRINT CSR(X-M,Y+M);CHR$(96)
580 PRINT CSR(X-1-M,Y);CHR$(96)
590 PRINT CSR(X-M,Y-M);CHR$(96)
600 PRINT CHR$(8)
610 RETURN
999 END
```

プログラムの概要

第6－2表にプログラムの概略フローを示します。さして広くもないテレビ画面をできるだけ豪華にみせるため，16行の画面の下3行に仕掛け花火，上13行の空間に打上げ花火を表示させます。打上げ花火の場合は，花火の表示と同時に効果音を出力します（テレビのボリュームを上げておくと効果が大きい？）。

このプログラムの主役は何といっても乱数。規則的に上がる花火では見ている方があきてしまいます。乱数によって，花火の色，打上げる高さ，打上げる地点，タイミングを多様に変化させます。

プログラムリストおよび表示画面例は写真のとおりです。

なおこのプログラムは関東バイトショップ，山下さんの力作です。

7 リアルタイム・ゲーム

SNAKE&MICE

TK80BS機械語

吉田紀彦

毎年5月に東京大学では五月祭が開催されていますが，私の所属する学科の'78の五月祭企画の一つ，「マイコンを使ったTVゲーム」のためにプログラミングしたのがこの「へびとねずみゲーム」です。

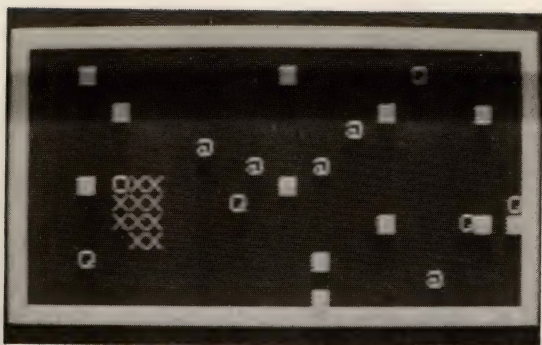
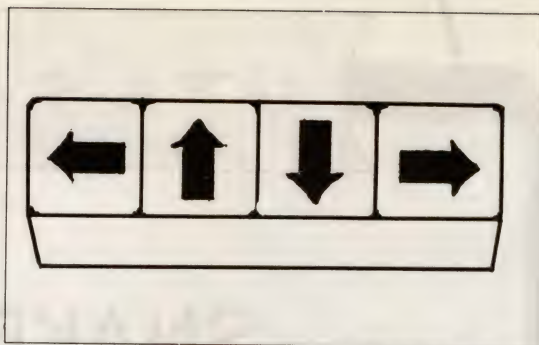
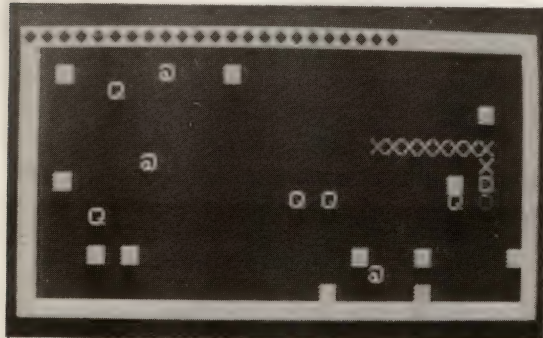
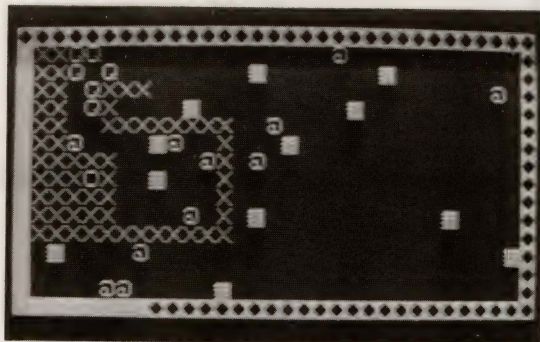
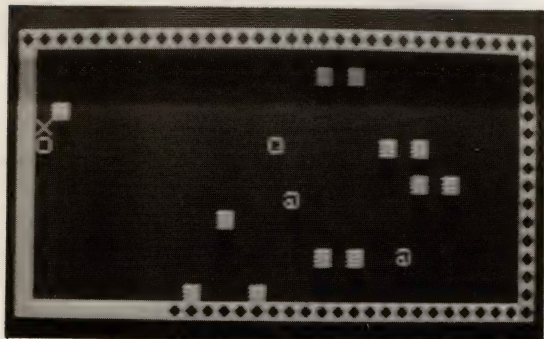
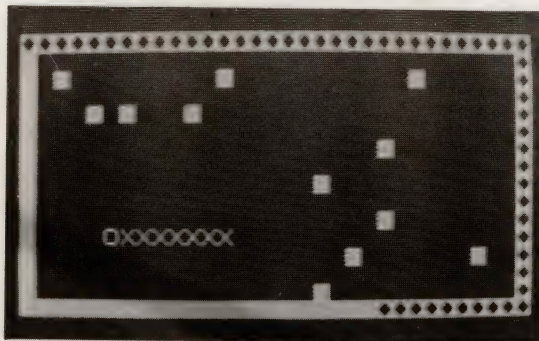
自分で楽しむのではなく多くの人に楽しんでもらうためのゲームには，やっておもしろいことはいうまでもなく，他人がやっているのを見ても面白いこと，長くても数分以内に決着がつくことなどの条件が要求されます。そこで，雑誌などに公表されているゲーム・プログまムをそのまま流用するのでは能がないということで，数人でプロジェクト・チームを組み，いろいろと適当なゲームのアイデアを考えそして探した結果，この「(略して）へびゲーム」を含む幾つかをやることになりました。

「へびゲーム」のもとになっているのは，東京都調布市にある電気通信大学で'77の11月に行なわれた調布祭の時に，「MMA（マイクロコンピュータを作る会)」という団体がデモンストレーションの一つとして，拡張したLkit—8を使ってやっていたゲームです。それを見た時の記憶をもとに，多少アレンジした上で，TK-80BS用に独自にプログラミングしたわけで，従って，ゲームのアイデアそのものはオリジナルではありませんが，少なくともプログラムに関する限り，すべての責任は私にあります。

《第7—1図》用いられる記号

——ゲームのやり方——

キー・ボードを操作してへびを動かしていき，走り回るねずみをつかまえて全部たべてしまえば良いのです。とはいっても，画面上を本物のへびやねずみが動くわけにはいきませんから，第7-1

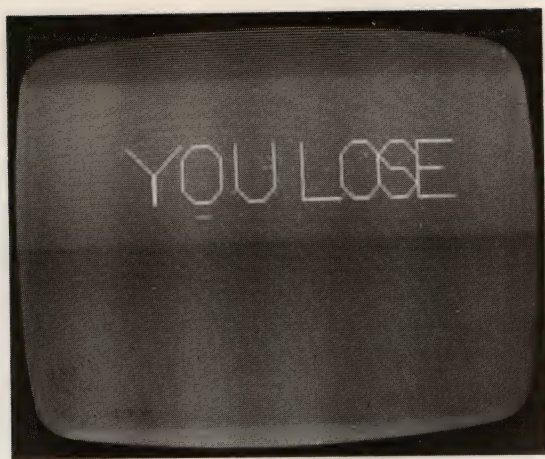
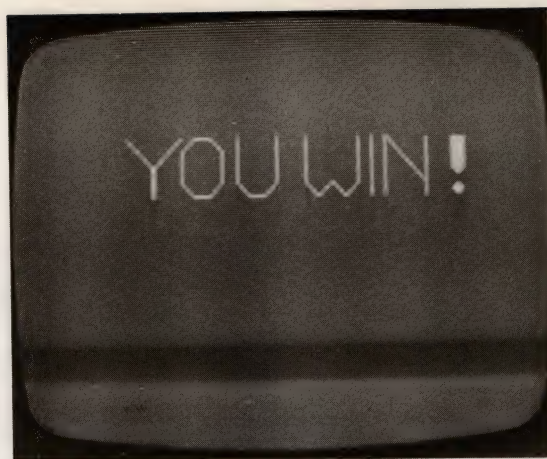
《写真7—1》タイトルの表示

〈写真7-2〉スペース・バーを押すと最初のねずみの位置が決定

〈写真7-3〉 キーボードにへびの移動方向を書いたプレートをつける

〈写真7-4〉〔3〕のキーを押すとへびは下の方向へ頭を向ける

〈写真7-5〉へびの頭がねずみにくっつくと，ねずみは食べられる

〈写真7-6〉へびが頭だけになってしまうと，飢え死にで敗け

〈写真7-7〉時間内にすべてのねずみを食べると，へびの勝ち

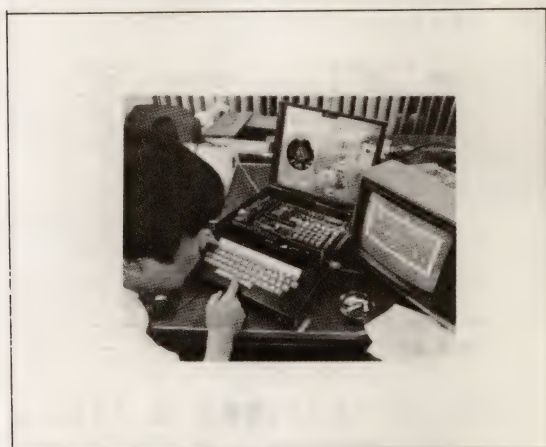
図のような記号を使って表わしています。ゲームのやり方を順に説明していきましょう。

1，画面に"SNAKE & MICE"とタイトルが表示されている時（**写真7-1**）にキー・ボード一番下にある横長のスペースバーを押すと，壁が描かれていき，そしてへびとねずみ，障害物が現われます(**写真7-2**)。時たま"SORRY……というメッセージがでて壁が描き直されることがあります。

2，もう一度スペースバーを押すと，画面上のへびとねずみが動き始めます。

3，ねずみの動きはデタラメですが，へびの動き方向はキーを押して指定することができます。1，2，3，4のキーが各々，左，上，下，右に対応していて，それ以外のキーは無視されます(もちろんBREAKとカナは別。展示の時には**写真7-3**のようなプレートを作って使いました)。進むべき方向が空白でない場合には，他のいずれかの方向をデタラメに選んで進みます。なお，一旦キーを押せば離した後も，へびは壁か障害物につかまるまで同じ方向に進み続けます。**写真7-4**はへびが右へ動いている途中で3（下）のキーを押したところです。

4，へびの頭がねずみにくっつくと，ねずみはへびに食べられてしまい（**写真7-5**），へびは胴体が一つ分だけ長くなります。

《写真7-8》敗けた場合，敗けのコメントが表示される

《写真7-9》勝った場合，勝ちのコメントが表示される

《写真7-10》オプショナルRAM2Kを実装すれば今後のプログラムにも使える

《写真7-11》熱中すると，おもしろくてなかなかやめられない

5. オスねずみとメスねずみとがぶつかると，子供を2匹生んで倍にふえることがあります。ぶつかった場所が#に変わり，大低その周辺にオス・メス2匹ずつが現れるようになっています。

6, ねずみの数が32匹を越えると，ふえすぎで敗けになります。へびがとぐろを巻いたなり壁と障害物との隙間はいり込んだりして頭の周囲に空白がなくなってしまうと動けなくなって敗けになります。

へびがねずみを食べずに動き回ってばかりいたり，急な曲り方をしたりすると，胴体が少しずつ短くなっていきます。へびが頭だけになってしまうと，飢え死にで敗け（**写真7-6**）。

一定の時間がたつごとに壁に一つずつ穴があいていきます。壁が穴だらけになってしまうと，時間切れで敗け。

7, 時間制限内にすべてのねずみを食べてしまえば，勝ちです（**写真7-7**）。

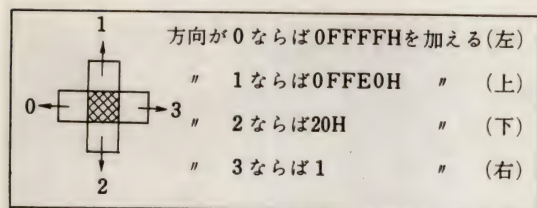
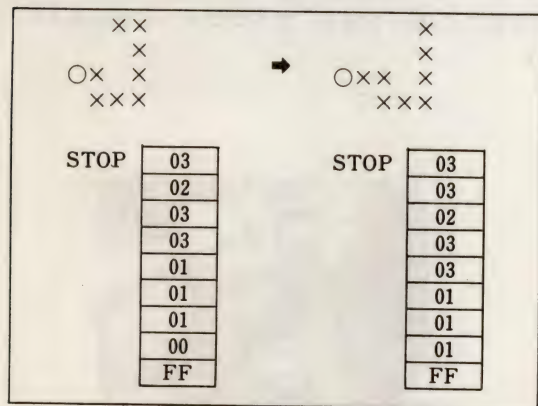
8. 勝つか敗けるかしてゲームが終了すると，"YOU WIN!"または"YOU LOSE"とメッセージが出て（**写真7-8，7-9**），1,に戻ります。

——プログラムの走らせ方——

システムとしては，TK—80BSを使い，ハードウエアの改造は一切していません。

プログラム＋ワークエリアが9000H番地から1.5Kバイト程，メッセージバッファが9C00H番地から1Kバイト程，スタックが9000H番地より下をそれぞれ占めています。従って，あらかじめオプショナルRAM2Kバイトを実装したTK-80BSなら，プログラムをそのままの形で走

B	一時退避用レジスタ、カウンタ
C	カウンタ
DE	アドレス・ポインタ
HL	アドレス・ポインタ、乱数

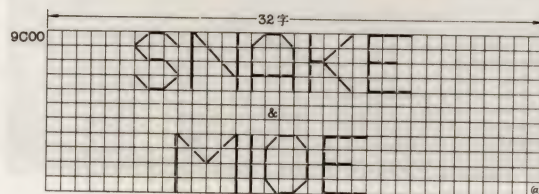
《第7-2図》各レジスタの用途

《第7-3図》移動方向を決めるアドレス

《第7-4図》へびの形を記憶するバッファ

らせることができます(**写真7-10, 7-11**)。実行開始番地は9000Hです。オプショナルRAMは今後の事を考えればこの機会にそなえてもよいでしょう。現在4個で8,800円程度です。

その他のシステムで走らせるためのアドバイスを少ししますと，まず基本構成のBSの場合には二つの方法があります。一つは，メッセージバッファを適当な空き番地に移し，9006H, 908AH, 9090H, 90FFH番地にある命令のオペレンドをそれぞれ書き換える。もう一つは，メモリを買って来る。2114が二つですから5.000円あれば間に合います。これ以外の8080系のシステムの場合には，ユーザーズ・エリアの番地がずれていたりI/Oルーチンが違っていたりでかなりめんどうになりますか，特にビデオRAMそのものをワークエリアとして使っているという点に注意してください。

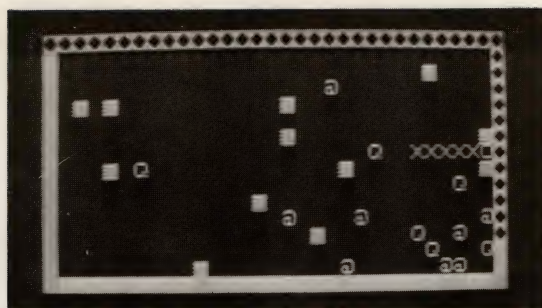
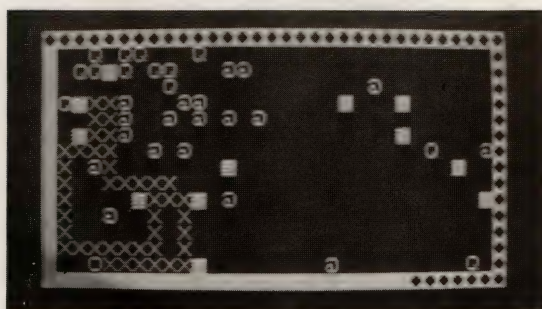
――プログラムについて――

"1"	⟷	0
"2"	⟷	1
"3"	⟷	2
"4"	⟷	3
"␣"	⟷	0EFH
入力なし	⟷	0FFH

《第7-5図》キー入力ルーチンと入力部分対応図

《第7-6図》メッセージの作り方

プログラム・リストと主なルーチンの簡単なフローチャートを示しておきました。これらを見ればだいたい理解して頂けると思います。

サブルーチン・コールをかなり多用したプログラムになっていますが，これは，このような作り方をすると全体が非常に見やすくなり，デバックが簡単にできるからです。ただし，各レジスタの用途をあらかじめ決めておかないと混乱して収捨がつかなくなるので，このプログラムでは一応**第7-2図**のように決めております。各サブルーチンの間などにNOP (00) が三つずつ挿入されているのは，プログラムを改造する時にジャンプ命令を入れたりして使えるようにしてあるのです。

画面上をいろいろな記号が縦横に走り回るというゲームなので，画面上のある位置に対して上下左右いずれかの方向に隣接している位置のアドレスを計算するという手続きが基本になります。1行の文字数が32 (20H) ですから，ある一定の定数を中心となっている位置のアドレスに加えれば求めるべきアドレスが得られるわけですが，直接それらの定数で方向を指定するのでは不便なので，このプログラムでは0~3のい

《写真7-12》障害物の間に頭をつっこんだへび

《写真7-13》へびがねずみを食べる速度とねずみが子供を生む量のバランスがとれれば，ゲームはどんどんできます。この場面は，へびの頭がどんどん進んでいくと，どこにもいけなくなるのでへびは死への道を歩んでいます。

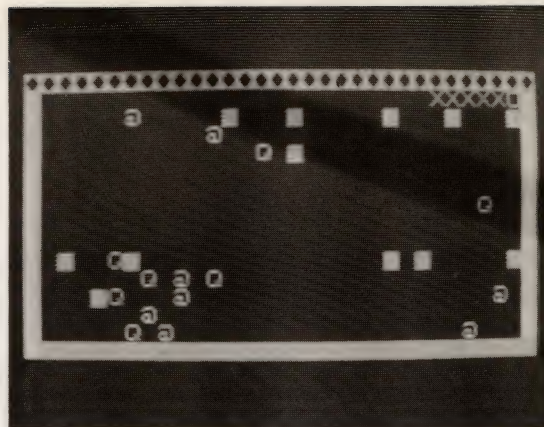
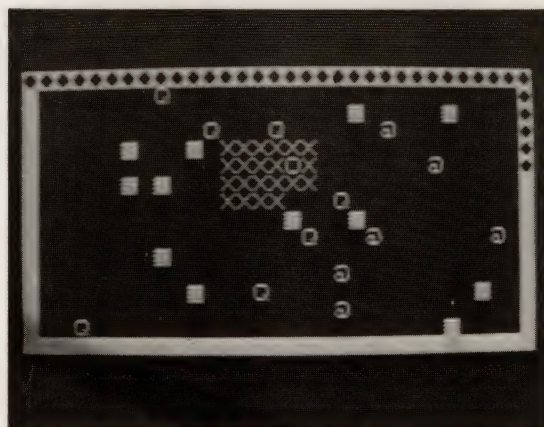
ずれかで方向を指定すれば**第7-3図**に従って対応するアドレスが計算されるようになっています。

そこで，コメントなどで「方向」と書いてある場合には，この0～3の数を指していると考えてください。

なお，ラベルSTOPに続く数+バイトはへびの形を記憶しておくためのバッファで，次の“╳”がどの方向にあるかが順々に収められ，末尾にテエンドマーク（0FFH）がついています（**第7-4図**）。

第7－5図は，キー入力ルーチンKYSCによって変換されるコードと入力記号との対応関係です。

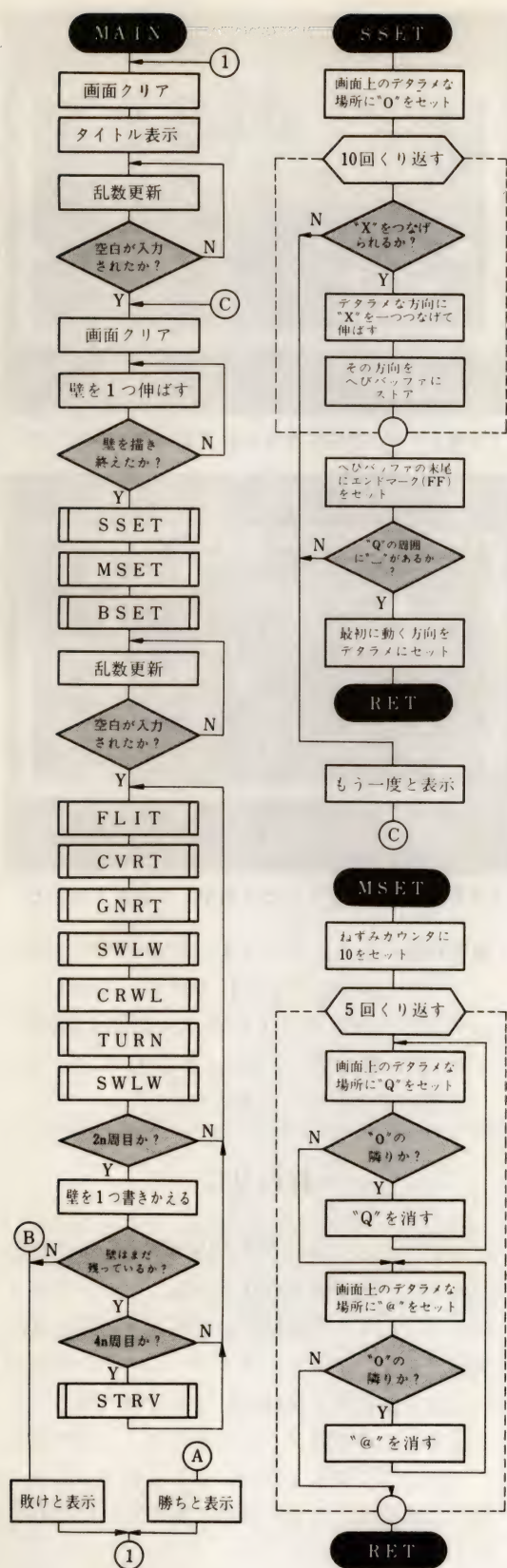
――メッセージについて――

五月祭での展示の時には，TK-80BSで使えるグラフィック・シンボルを組み合わせて文字を作り，写真のようなメッセージを表示させました。これらのメッセージの作り方を図解すると，

《写真7-14》方向変換を忘れて死んだドジなへび

《写真7-15》食べることに熱中しすぎたドジヘビ

第7-6図のようになります（最後の“ⓐ”＝00はエンドマークで，表示されません)。しかし，この例にこだわらず，“CONGRATULATION!”とか“アンタ　ノ　カチ!”とか独自のメッセージを考えるのも良いと思います。

――終わりに――

このゲームは，プログラム内のいくつかの定数を変えて，難易度を変化させることができます。例えば，ねずみの初期値が2匹違うと難易度は大きく変化しますし，タイマ・ルーチン内の時間定数を減らすと全体的に速くなって難しくなります。このプログラムのままだと，最初はかなりまごつくでしょうが，5～6回やってキー操作に慣れてくると，勝率7割くらいになれます。

また，機械語で入力する量が多いので，テーズに入れておくとよいでしょう。この時，一度

に全部入れてやろうと考えずに，たとえば，9000H～9100Hまでとりあえず入れて，次の日に9100H～9200Hに入力するというようにして，カセットに入れるとよいでしょう。

最近はBASICによるゲームが流行しているようで，確かに凝ったメッセージをいろいろと出したい時とか数値演算が必要な時には，BASICもそれなりの威力は発揮するでしょう。しかし，インタプリタ方式ですからスピードも遅いし会話処理しかできないので，画面の動きを楽しむようなゲームを作るのはかなり難しいのではないかと思います。以前，試しにBSのLevel-IBASICでライフゲームを作ったところ，世代がかわるのに数分もかかったほどですから。

ついでに書いておきますと，BSのLevel-1BASICインタプリタではCALL文で呼ばれた機械語サブルーチンの中でDEレジスタの内容を破壊してしまうと，正常な動作が保証されなくなるので注意してください。

MAIN
画面クリア
タイトル表示
乱数更新
空白が入力されたか？
画面クリア
壁を1つ伸ばす
壁を描き終えたか？
SSET
MSET
BSET
乱数更新
空白が入力されたか？
FLIT
CVRT
GNRT
SWLW
CRWL
TURN
SWLW
2n周目か？
壁を1つ書きかえる
壁はまだ残っているか？
4n周目か？
STRV
負けと表示
勝ちと表示

SSET
画面上のデタラメな場所に"O"をセット
10回くり返す
"X"をつなげられるか？
デタラメな方向に"X"を一つつなげて伸ばす
その方向をへびバッファにストア
へびバッファの末尾にエンドマーク(FF)をセット
"Q"の周囲に"＿"があるか？
最初に動く方向をデタラメにセット
RET
もう一度と表示

MSET
ねずみカウンタに10をセット
5回くり返す
画面上のデタラメな場所に"Q"をセット
"O"の隣りか？
"Q"を消す
画面上のデタラメな場所に"@"をセット
"O"の隣りか？
"@"を消す
RET

フローチャートⒶ

フローチャートⒷ

FLIT
ビデオRAMを
スキャンしていく
"Q"か？
周囲に
"＿"があるか
？
デタラメな方向に
1つ動かし"C"
に書きかえる
"@"か？
周囲に
"＿"があるか
？
デタラメな方向に
1つ動かし"G"
に書きかえる
RET
GNRT
ビデオRAMを
スキャンしていく
"Q"か？
周囲に
"@"があるか
？
RET
ねずみカウンタを
2回デクリメント
"#"を表示
"#"の隣
4ヶ所について
くり返す
"#"の隣は
"＿"か？
その隣に"Q"か
"@"を書き、外側
に1つ動かす
ねずみカウンタを
インクリメント
"#"を消す
〔ねずみカウンタ〕
<32？
B
RET
CRWL
"O"の周囲
に"＿"があるか
？
B
方向指定キー
が押されたか
？
方向レジスタにストア
方向レジスタからロード
その方向は
"＿"か？
方向をデタラメにセット
"Q"をその方向に
1つ進める
へびバッファを
スキャンしていく
エンドマーク
(FF)か？
へびバッファの内容
を1つ後ろにずらす
次の"X"のアドレス
を求める
胴体末尾の"X"
を消す
RET
TURN
へびバッファを
初めから3つ
スキャンする
エンドマーク
か？
次の"X"のアドレス
を求める
3番目の"X"の
アドレスをストア
"O"の隣
4ヶ所について
くり返す
隣のアドレス＝
ストアしたアドレス
？
RET
STRV
STRV
へびバッファを
スキャンしていく
エンドマーク
か？
次の"X"のアドレス
を求める
エンドマークを
1つ前へずらす
胴体末尾の"X"
を消す
へびバッファ
の内容はエンドマーク
だけか？
B
RET

フローチャート©

へびとねずみプログラムリスト

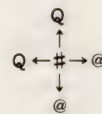
```
*   SNAKE & MICE GAME  *
*            /REVISED/ *
*                      *
*   CODED BY N YOSHIDA *
          *
TTL     EQU   9C00H  }
SRY     EQU   9DC00H } メッセージバッファの先頭番地
YWN     EQU   9E00H  }
YLS     EQU   9F00H  }
VTOP    EQU   7E00H     ビデオRAMの先頭番地
VSZE    EQU   200H      ビデオRAMのバイト数
          *
```

アドレス	機械語	ラベル	ニーモニック		備考
9000	310090	INTL	LXI	SP,INTL	
3	CD2994		CALL	CLR	
6	21009C		LXI	H,TTL	
9	CD3F94		CALL	MSG	
900C	CDF193	STRT	CALL	RNDM	タイトルを表示
F	CD5194		CALL	KYSC	空白が入力されるまで乱数を更新し
2	FEEF		CPI	0EFH	つつ待つ
4	C20C90		JNZ	STRT	
9017	310090	WSET	LXI	SP,INTL	
A	CD2994		CALL	CLR	壁をセットするルーティン
D	3E5C		MVI	A,5CH	
F	320695		STA	WCNT	画面外周の文字数は92
22	F5		PUSH	PSW	
3	21007E		LXI	H,VTOP	
6	220295		SHLD	WADR	
9	E5		PUSH	H	
A	0680		MVI	B,80H	
902C	CD1594	WS1	CALL	TIMB	文字コード"■"
F	CD7F91		CALL	WALL	
32	C22C90		JNZ	WS1	
5	E1		POP	H	画面を一周するまでくり返す
6	F1		POP	PSW	
7	320695		STA	WCNT	
A	220295		SHLD	WADR	
D	CDA590		CALL	SSET	
40	CD0B91		CALL	MSET	
3	CD5491.C		CALL	BSET	
9046	CDF193	WAIT	CALL	RNDM	空白が入力されるまで乱数を更新し
9	CD5194		CALL	KYSC	つつ待つ
C	FEEF		CPI	0EFH	
E	C24690		JNZ	WAIT	
51	000000				
9054	0E02	LOOP	MVI	C,2	
9056	C5	LO1	PUSH	B	
7	CDBA91		CALL	FLIT	
A	CD0F92		CALL	CVRT	
D	CD3192		CALL	GNRT	
60	CD1294		CALL	TIMC	
3	CD4D93		CALL	SWLW	
6	CDA192		CALL	CRWL	
9	CDF592		CALL	TURN	
C	CD1294		CALL	TIMC	
F	CD4D93		CALL	SWLW	
72	C1		POP	B	
3	0D		DCR	C	2サイクルに1つ壁タイマを進める
4	C25690		JNZ	LO1	
7	06CB		MVI	B,0CBH	文字コード"◘"
9	CD7F91		CALL	WALL	
C	CA9090		JZ	LSEM	時間切れで負け
F	E603		ANI	3	8サイクルに1つへびを短くする
81	CC2693		CZ	STRV	
4	C35490		JMP	LOOP	
7	000000				

908A	21009E	WINW	LXI	H,YWN	"YOU␣WIN!" と表示
D	C39390		JMP	LS1	
9090	21009F	LSEM	LXI	H,YLS	"YOU␣LOSE" と表示
9093	CD0694	LS1	CALL	TIMG	
6	CD2994		CALL	CLR	
9	CD3F94		CALL	MSG	
C	CD0994		CALL	TIMF	
F	C30090		JMP	INTL	
A2	000000				
90A5	CDF193	SSET	CALL	RNDM	へびの頭をセットするルーティン
8	7C		MOV	A,H	
9	E601		ANI	1	
B	67		MOV	H,A	
C	11007E		LXI	D,VTOP	
F	19		DAD	D	画面上のデタラメな位置→HL
B0	7E		MOV	A,M	
1	FE20		CPI	20H	文字コード "␣(空白)"
3	C2A590		JNZ	SSET	そこが "␣" でなければもう一度文
6	360F		MVI	M,0FH	字コード "O"
8	220895		SHLD	SADR	
B	EB		XCHG		
C	0E0A		MVI	C,0AH	へびの胴体の長さは10
E	210B95		LXI	H,STOP	
90C1	E5	SS1	PUSH	H	へびの頭に胴体をつなげていくルー
2	EB		XCHG		ティン
3	3E20		MVI	A,20H	
5	CDD693		CALL	SRCH	
9 8	C2FC90		JNZ	SS3	HLの周囲に "␣" がなければ
90CB	CDF193	SS2	CALL	RNDM	
E	7D		MOV	A,L	
F	E603		ANI	3	
D1	47		MOV	B,A	
2	CDB593		CALL	DRCT	
5	7E		MOV	A,M	
6	FE20		CPI	20H	
8	C2CB90		JNZ	SS2	そこが "␣" でなければもう一度文
B	3618		MVI	M,18H	字コード "X"
D	EB		XCHG		
E	E1		POP	H	
F	70		MOV	M,B	
E0	23		INX	H	
1	0D		DCR	C	
2	C2C190		JNZ	SS1	
6	36FF		MVI	M,0FFH	
7	3E20		MVI	A,20H	
9	2A0895		LHLD	SADR	
C	CDD693		CALL	SRCH	
F	C2FC90		JNZ	SS3	へびの頭の周囲に "␣" がなければ
F2	CDF193		CALL	RNDM	
5	7D		MOV	A,L	
5	E603		ANI	3	
8	320A95		STA	SFLG	最初に動く方向をセット
B	C9		RET		
90FC	CD2994	SS3	CALL	CLR	"SORRY.." と表示してやり直し
F	21C09D		LXI	H,SRY	
02	CD3F94		CALL	MSG	
5	C31790		JMP	WSET	
8	000000				
910B	3E05	MSET	MVI	A,5	ねずみはオス・メス5 匹ずつ
D	47		MOV	B,A	
E	87		ADD	A	2倍する
F	320795		STA	MCNT	
9112	CDF193	MS1	CALL	RNDM	オスねずみをセットするルーティン
5	7C		MOV	A,H	
6	E601		ANI	1	
8	67		MOV	H,A	
9	11007E		LXI	D,VTOP	
C	19		DAD	D	
D	7E		MOV	A,M	
E	FE20		CPI	20H	
20	C21291		JNZ	MS1	そこが "␣" でなければもう一度文
3	3E11		MVI	A,11H	字コード "O"
5	77		MOV	M,A	
6	2A0895		LHLD	SADR	

9	CDD693		CALL	SRCH	
C	CA1291		JZ	MS1	へびの頭の隣なら消してもう一度メ
912F	CDF193	MS2	CALL	RNDM	スねずみをセットするルーティン
32	7C		MOV	A,H	
3	E601		ANI	1	
5	67		MOV	H,A	
6	11007E		LXI	D,VTOP	
9	19		DAD	D	
A	7E		MOV	A,M	
B	FE20		CPI	20H	
D	C22F91		JNZ	MS2	
40	3E00		MVI	A,0	文字コード〝@〞
2	77		MOV	M,A	
3	2A0895		LHLD	SADR	
6	CDD693		CALL	SRCH	
9	CA2F91		JZ	MS2	
C	05		DCR	B	
D	C21291		JNZ	MS1	
50	C9		RET		
1	000000				
9154	060C	BSET	MVI	B,0CH	障害物は12個
9156	CDF193	BS1	CALL	RNDM	障害物をセットするルーティン
9	7C		MOV	A,H	
A	E601		ANI	1	
C	67		MOV	H,A	
D	7D		MOV	A,L	
E	E6DE		ANI	0DEH	奇数行、奇数列に制限
60	6F		MOV	L,A	
1	11007E		LXI	D,VTOP	
4	19		DAD	D	
5	7E		MOV	A,M	
6	FE20		CPI	20H	
8	C25691		JNZ	BS1	
B	3ECD		MVI	A,0CDH	文字コード〝▦〞
D	77		MOV	M,A	
E	2A0895		LHLD	SADR	
71	CDD693		CALL	SRCH	
4	CA5691		JZ	BS1	へびの頭の隣なら消してもう一度
7	05		DCR	B	
8	C25691		JNZ	BS1	
B	C9		RET		
C	000000				
917F	2A0495	WALL	LHLD	WDAD	壁を1つ伸ばすルーティン
82	3A0695		LDA	WCNT	
5	FE5C		CPI	5CH	
7	C28D91		JNZ	WA1	
A	210100		LXI	H,1	左上隅ならば方向を右に変える
918D	FE3D	WA1	CPI	3DH	
F	C29591		JNZ	WA2	
92	212000		LXI	H,20H	右上隅ならば方向を下に変える
9195	FE2E	WA2	CPI	2EH	
7	C29D91		JNZ	WA3	
A	21FFFF		LXI	H,0FFFFH	右下隅ならば方向を左に変える
919D	FE0F	WA3	CPI	0FH	
F	C2A591		JNZ	WA4	
A2	21E0FF		LXI	H,0FFE0H	左下隅ならば方向を上に変える
91A5	EB	WA4	XCHG		
6	2A0295		LHLD	WADR	
9	70		MOV	M,B	壁を1つ伸ばす
A	19		DAD	D	
B	3D		DCR	A	壁を描き終えたらZ＝1
C	320695		STA	WCNT	
F	220295		SHLD	WADR	
B2	EB		XCHG		
3	220495		SHLD	WDAD	
6	C9		RET		
7	000000				
91BA	010002	FLIT	LXI	B,VSZE	それぞれのねずみを上下左右に1つ
D	21007E		LXI	H,VTOP	動かすルーティン
91C0	7E	FL1	MOV	A,M	
1	FE11		CPI	11H	オスねずみ〝O〞か？
3	C2E191		JNZ	FL4	

6	3E20		MVI	A,20H	
8	CDD693		CALL	SRCH	
B	C2E091		JNZ	FL3	周囲に〝␣〟がなければ動かさない
91CE	CDF193	FL2	CALL	RNDM	
D1	7D		MOV	A,L	
2	E603		ANI	3	
4	CDB593		CALL	DRCT	
7	7E		MOV	A,M	
8	FE20		CPI	20H	動く方向が〝␣〟でなければもう一
A	C2CE91		JNZ	FL2	度
D	12		STAX	D	〝␣〟→[DE]
E	3603		MVI	M,3	〝C〟→[HL]
91E0	EB	FL3	XCHG		
91E1	7E	FL4	MOV	A,M	
2	FE00		CPI	0	メスねずみ〝@〟か?
4	C20292		JNZ	FL7	
7	3E20		MVI	A,20H	
9	CDD693		CALL	SRCH	
C	C20192		JNZ	FL6	
91EF	CDF193	FL5	CALL	RNDM	
F2	7D		MOV	A,L	
3	E603		ANI	3	
5	CDB593		CALL	DRCT	
8	7E		MOV	A,M	
9	FE20		CPI	20H	
B	C2EF91		JNZ	FL5	
E	12		STAX	D	〝␣〟→[DE]
F	3607		MVI	M,7	〝G〟→[HL]
9201	EB	FL6	XCHG		
9202	23	FL7	INX	H	
3	0D		DCR	C	
4	C2C091		JNZ	FL1	
7	05		DCR	B	
8	C2C091		JNZ	FL1	
B	C9		RET		
C	000000				
920F	010002	CVRT	LXI	B,VSZE	変換した記号を元に戻すルーティン
12	21007E		LXI	H,VTOP	
9215	7E	CV1	MOV	A,M	
6	FE03		CPI	3	
8	C21D92		JNZ	CV2	
B	3611		MVI	M,11H	〝C〟を〝Q〟に戻す
921D	FE07	CV2	CPI	7	
F	C22492		JNZ	CV3	
22	3600		MVI	M,0	〝G〟を〝@〟に戻す
9224	23	CV3	INX	H	
5	0D		DCR	C	
6	C21592		JNZ	CV1	
9	05		DCR	B	
A	C21592		JNZ	CV1	
D	C9		RET		
E	000000				
9231	3A0795	GNRT	LDA	MCNT	オスねずみとメスねずみがぶつかっ
4	47		MOV	B,A	たら子供を生むルーティン
5	110002		LXI	D,VSZE	
8	21007E		LXI	H,VTOP	
923B	7E	GN1	MOV	A,M	
C	FE11		CPI	11H	
E	C24D92		JNZ	GN2	
41	D5		PUSH	D	
2	E5		PUSH	H	
3	3E00		MVI	A,0	
5	CDD693		CALL	SRCH	〝Q〟の周囲に〝@〟がいたらそれ
8	CA5792		JZ	GN3	を消してジャンプ
B	E1		POP	H	
C	D1		POP	D	
924D	23	GN2	INX	H	
E	1D		DCR	E	
F	C23B92		JNZ	GN1	
52	15		DCR	D	
3	C23B92		JNZ	GN1	
6	C9		RET		

9257	05	GN3	DCR	B	ねずみカウンタを2つ減らす
8	05		DCR	B	
9	3E23		MVI	A,23H	“Q”の場所に“#”を表示
B	12		STAX	D	
C	CD1294		CALL	TIMC	
F	0E03		MVI	C,3	
9261	79	GN4	MOV	A,C	“#”の周囲の“␣”を“Q”か
2	CDB593		CALL	DRCT	“@”で置き換えてさらに1つ外側
5	7E		MOV	A,M	に動かす
6	FE20		CPI	20H	
8	C28892		JNZ	GN7	Q
B	3600		MVI	M,0	↑
D	79		MOV	A,C	Q←#→@
E	FE02		CPI	2	↓
70	D27592		JNC	GN5	@
3	3611		MVI	M,11H	
9275	04	GN5	INR	B	ねずみカウンタを1つふやす
6	D5		PUSH	D	
7	EB		XCHG		
8	79		MOV	A,C	
9	CDB593		CALL	DRCT	
C	7E		MOV	A,M	
D	FE20		CPI	20H	
F	C28792		JNZ	GN6	
82	1A		LDAX	D	
3	77		MOV	M,A	
4	3E20		MVI	A,20H	
6	12		STAX	D	
9287	D1	GN6	POP	D	
9288	0D	GN7	DCR	C	
9	F26192		JP	GN4	
C	CD1294		CALL	TIMC	
F	3E20		MVI	A,20H	
91	12		STAX	D	“#”を消す
2	E1		POP	H	
3	D1		POP	D	
4	78		MOV	A,B	
5	FE20		CPI	20H	ねずみが32匹を越えたらふえすぎ
7	D29090		JNC	LSEM	で敗け
A	320795		STA	MCNT	
D	C9		RET		
E	000000				
92A1	2A0895	CRWL	LHLD	SADR	へびを這わせるルーティン
4	3E20		MVI	A.20H	
6	CDD693		CALL	SRCH	へびの頭の周囲に“␣”がなければ
9	C29090		JNZ	LSEM	動けなくなって敗け
C	CD5194		CALL	KYSC	
F	FE04		CPI	4	方向指示キーが押されていたらそれ
B1	D2B792		UNC	CR2	をSFLGにセット
92B4	320A95	CR1	STA	SFLG	
92B7	3A0A95	CR2	LDA	SFLG	
A	47		MOV	B,A	
B	CDB593		CALL	DRCT	
E	7E		MOV	A,M	
F	FE20		CPI	20H	
C1	CACD92		JZ	CR3	
4	CDF193		CALL	RNDM	動くべき方向が“␣”でなければ乱
7	7D		MOV	A,L	数で方向を決め直す
8	E603		ANI	3	
A	C3B492		JMP	CR1	
92CD	78	CR3	MOV	A,B	
E	2F		CMA		
F	E603		ANI	3	方向を反転する
D1	47		MOV	B,A	
2	360F		MVI	M,0FH	頭“O”を1つ進める
4	3E18		MVI	A,18H	
6	12		STAX	D	
7	220895		SHLD	SADR	へびの頭の位置を更新
A	210B95		LXI	H,STOP	
92DD	7E	CR4	MVO	A,M	へびバッファに−1が現われるまで
E	A7		ANA	A	その内容を1つずつ後ろにずらし胴
F	FAEE92		JM	CR5	体“X”の位置をDEでトレースし
E2	70		MOV	M,B	ていく
3	47		MOV	B,A	

4	E5		PUSH	H	
5	CDB593		CALL	DRCT	
8	EB		XCHG		
9	E1		POP	H	
A	23		INX	H	
B	C3DD92		JMP	CR4	
92EE	3E20	CR5	MVI	A,20H	
F0	12		STAX	D	胴体末尾の〝X″を消す
1	C9		RET		
2	000000				
92F5	2A0895	TURN	LHLD	SADR	急な曲り方をしたかどうか判断するルーティン
8	EB		XCHG		
9	D5		PUSH	D	
A	210B95		LXI	H,STOP	
D	0E03		MVI	C,3	
92FF	7E	TU1	MOV	A,M	
00	A7		ANA	A	
1	FA2193		JM	TU3	胴体の長さが2以下ならリターン
4	E5		PUSH	H	
5	CDB593		CALL	DRCT	
8	EB		XCHG		
9	E1		POP	H	
A	23		INX	H	
B	0D		DCR	C	
C	C2FF92		JNZ	TU1	
F	43		MOV	B,E	3番目の〝X″の位置の下位バイト→B
10	D1		POP	D	
1	0E03		MVI	C,3	
9313	79	TU2	MOV	A,C	
4	CDB593		CALL	DRCT	
7	7D		MOV	A,L	
8	B8		CMP	B	3番目の〝X″が頭の隣にあったら胴体を1つ短くする
9	CA2693		JZ	STRV	
C	0D		DCR	C	
D	F21393		JP	TU2	XXXXX
20	C9		RET		XO
9321	D1	TU3	POP	D	
2	C9		RET		
3	000000				
9326	2A0895	STRV	LHLD	SADR	へびを短くするルーティン
9	EB		XCHG		
A	210B95		LXI	H,STOP	
932D	7E	ST1	MOV	A,M	
E	A7		ANA	A	
F	FA3C93		JM	ST2	
32	E5		PUSH	H	
3	CDB593		CALL	DRCT	胴体の位置をDEでトレースしていく
6	EB		XCHG		
7	E1		POP	H	
8	23		INX	H	
9	C32D93		JMP	ST1	
933C	2B	ST2	DCX	H	へびバファのエンドマークを1つ前にずらす
D	36FF		MVI	M,OFFH	
F	3E20		MVI	A,20H	
41	12		STAX	D	胴体を1つ短くする
2	3A0B95		LDA	STOP	
5	A7		ANA	A	へびが頭だけになってしまったら飢え死にで敗け
6	FA9090		JM	LSEM	
9	C9		RET		
A	000000				
934D	2A0895	SWLW	LHLD	SADR	へびがねずみを食べるルーティン
50	3E11		MVI	A,11H	
2	4F		MOV	C,A	
3	CDD693		CALL	SRCH	へびの頭の周囲にオスねずみがいたらそれを消してSW1へ
6	CA6193		JZ	SW1	
9	EB		XCHG		
A	3E00		MVI	A,0	
C	4F		MOV	C,A	
D	CDD693		CALL	SRCH	へびの頭の周囲にメスねずみがいたらそれを消してSW1へ
60	C0		RNZ		
9361	210B95	SW1	LXI	H,STOP	
9364	7E	SW2	MOV	A,M	
5	A7		ANA	A	
6	FA7A93		JM	SW3	
9	47		MOV	B,A	方向を憶えておく

A	E5		PUSH	H	
B	CDB593		CALL	DRCT	
E	71		MOV	M,C	˝X″をQ″か˝@″で置きかえる
F	CD1294		CALL	TIMC	˝X″に戻す
72	3618		MVI	M,18H	
4	EB		XCHG		
5	E1		POP	H	
6	23		INX	H	
7	C36493		JMP	SW2	
937A	E5	SW3	PUSH	H	
B	EB		XCHG		
C	3E20		MVI	A,20H	
E	CDD693		CALL	SRCH	
81	E1		POP	H	胴体末尾の周囲に˝␣″がなければ
2	C2A593		JNZ	SW6	胴体は伸ばさない
5	78		MOV	A,B	同じ方向に伸ばす
6	E5		PUSH	H	
9387	CDB593	SW4	CALL	DRCT	
A	7E		MOV	A,M	
B	FE20		CPI	20H	
D	CA9A93		JZ	SW5	
90	CDF193		CALL	RNDM	伸ばすべき方向が˝␣″でなければ
3	7D		MOV	A,L	乱数で方向を決め直す
4	E603		ANI	3	
6	47		MOV	B,A	
7	C38793		JMP	SW4	
939A	71	SW5	MOV	M,C	
B	CD1294		CALL	TIMC	
E	3618		MVI	M,18H	胴体を1つ伸ばす
0A0	E1		POP	H	
1	70		MOV	M,B	方向をへびバッファにセット
2	23		INX	H	へびバッファのエンドマークを1つ
3	36FF		MVI	M,0FFH	後ろにずらす
93A5	3A0795	SW6	LDA	MCNT	
8	3D		DCR	A	ねずみカウンタを1つ減らす
9	CA8A90		JZ	WINM	ねずみを食べ尽せば勝ち
C	320795		STA	MCNT	
F	C34D93		JMP	SWLW	
B2	000000				
93B5	A7	DRCT	ANA	A	DEで指定された位置に対してAで
6	C2BC93		JNZ	DR1	指定された側の隣の位置をHLに入
9	21FFFF		LXI	H,0FFFFH	れるルーティン
93BC	3D	DR1	DCR	A	[A]と方向との対応は
D	C2C393		JNZ	DR2	
C0	21E0FF		LXI	H,0FFE0H	01
93C3	3D	DR2	DCR	A	↑
4	C2CA93		JNZ	DR3	00←[DE]→11
7	212000		LXI	H,20H	↓
93CA	3D	DR3	DCR	A	10
B	C2D193		JNZ	DR4	
E	210100		LXI	H,1	
93D1	19	DR4	DAD	D	
2	C9		RET		
3	000000				
93D6	C5	SRCH	PUSH	B	HLで指定された位置の隣にAで指
7	47		MOV	B,A	定された記号があればその記号を画
8	EB		XCHG		面から消してZ=1、なければZ=
9	0E03		MVI	C,3	0にするルーティン
93DB	79	SR1	MOV	A,C	
C	CDB593		CALL	DRCT	
F	7E		MOV	A,M	
E0	B8		CMP	B	
1	CAEA93		JZ	SR2	
4	0D		DCR	C	
5	F2DB93		JP	SR1	
8	C1		POP	B	
9	C9		RET		
93EA	3620	SR2	MVI	M,20H	
C	C1		POP	B	
D	C9		RET		
E	000000				
93F1	2A0095	RNDM	LHLD	RSTR	乱数発生ルーティン
4	7C		MOV	A,H	[LH]を257倍して3を加える
5	85		ADD	L	
6	6F		MOV	L,A	

7	7C		MOV	A,H	
8	C603		ADI	3	
A	67		MOV	H,A	
B	7D		MOV	A,L	
C	CE00		ACI	0	
E	6F		MOV	L,A	
F	220095		SHLD	RSTR	
02	C9		RET		
3	000000				
9406	CD0994	TIMG	CALL	TIMF	タイマ・ルーティン群
9409	CD0C94	TIMF	CALL	TIME	
940C	CD0F94	TIME	CALL	TIMD	
940F	CD1294	TIMD	CALL	TIMC	
9412	CD1594	TIMC	CALL	TIMB	
9415	CD1894	TIMB	CALL	TIMA	
9418	E5	TIMA	PUSH	H	
9	210008		LXI	H,800H	
941C	2D	TI1	DCR	L	
D	C21C94		JNZ	TI1	
20	25		DCR	H	
1	C21C94		JNZ	TI1	
4	E1		POP	H	
5	C9		RET		
6	000000				
9429	3E20	CLR	MVI	A,20H	画面をクリアするルーティン
B	010002		LXI	B,VSZE	
E	11007E		LXI	D,VTOP	
9431	12	CL1	STAX	D	
2	13		INX	D	
3	0D		DCR	C	
4	C23194		JNZ	CL1	
7	05		DCR	B	
8	C23194		JNZ	CL1	
B	C9		RET		
C	000000				
943F	11607E	MSG	LXI	D,VTOP+60H	HLで指定されたメッセージバッファの内容をエンドマーク"@"が現れるまでビデオRAMに転送していくルーティン
9442	7E	MS	MOV	A,M	
3	A7		ANA	A	
4	C8		RZ		
5	12		STAX	D	
6	13		INX	D	
7	23		INX	H	
8	CD1894		CALL	TIMA	
B	C34294		JMP	MS	
E	000000				
9451	3AFE7D	KYSC	LDA	7DFEH	キー入力フラグを調べる
4	E620		ANI	20H	
6	CA5E94		JZ	KY1	
9	3AFC7D		LDA	7DFCH	入力キーの値をロードする
C	D630		SUI	30H	
945E	3D	KY1	DCR	A	
F	C9		RET		
			*		
			DS	2	
9500		RSTR	DS	2	乱数
9502		WADR	DS	2	壁についてのデータ
9504		WDAD	DS	1	
9506		WCNT	DS	1	
9507		MCNT	DS	2	ねずみの匹数
9508		SADR	DS	1	へびの頭の位置
950A		SFLG	DS	64	へびの動く方向
950B		STOP	END		へびの形(へびバファ)

8

BASICでリアルタイムに

ボーリング・ゲーム・プログラム

TK-80BSL I

野口武志

このボーリング・ゲームは未来のボーリング場をちょうど再現したものといえるでしょう。スコアも全部マイコンが付けてくれるので，あなたは投球に全神経をはらえばよいのです。

ゲームのやり方

1．プログラムを入力してRUNさせると，画面に順にタイトル，フレーム，ボール，スコア表それにレーンとピンが現われます。

2．レーンの下に（画面の右下）「Kヲ　オシナサイ」と表示されます。キー・ボードの「K」を押すとゲームが始まります。

3．レーンの下の左側にキャラクタ（DCH）が現われます。（人間のつもりです）

4．キー・ボード上の「J」，「K」，「L」で**3**.の人間を左右に動かしてください。人間はJで左へ動き，Lで右へ移動します。Kを押すと移動をやめて停止します。これで，ボールを投げる位置を決めるのですが，Kで停止させたあとも，位置を修正したい場合はJ，Lを押すことによって修正できます。

5．キー・ボード上の「I」を押すことにより人間の止っている位置からボールが投球されます。ボールは直進していますが，レーンの上でボールのコントロールができます。キー・ボード上のJを押すとボールは左へ曲がります。Lで右へ曲がり，Kで直進になります。本物のようにフックボールも可能です。

6．フレーム数，ボール数，ピンの状態，スコア表は自動的に進みますので，フレームが終わるごとに**2**. にもどってくり返してください。

《写真8—1》レーン，スコア表を表示

《写真8—2》Kを押すと人間が現われます

《写真8—3》キーで人間の投球位置を決定

《写真8—4》"I" で投球

《写真8—5》キーで球の移動を修正

《写真8—6》ピンがたおれる

《写真8—7》残った本数を表示

《写真8—8》二回目の投球

7．10フレームの第2投(スペア又はストライクの場合は3投目）が終わると「GAME OVER」の表示とともに，あなたの得点と評価が表示されます。なお，スコア表は表示されたままです。

8．もう一度ゲームをプレーする場合はYのキーを，やめる場合はNのキーを押せばSTOPします。

プログラムの説明

プログラムの概略を説明しておきます。

メイン・プログラム

行番号 **2〜39**：初めのタイトル，フレーム，ボール数，スコア表のわくと配列の初期値を設定します。

《写真8―9》ピンをたおす

*** ボ゛ーリング゛・ゲ゛ーム *** 0

4 フレーム　ボ゛ール 2

1	2	3	4	5
4 4 8	7 -	9 ◢	4 _	
6	7	8	9	10

ミス!!

《写真8―10》ミスの表示

アナタ ハ 90 テンデ゛ス
****
* GAME OVER *
****
モウ ヤメタホウガ゛ イイデ゛スヨ !!

1	2	3	4	5
4 4 8	7 - 15	9 ◢ 29	4 5 38	6 3 47
6	7	8	9	10
9 ◢ 62	5 2 69	6 1 76	6 3 85	5- 90

モウ イチト゛
シマスカ
Y ? N ?_

《写真8―11》スコアに得点を表示

1	2	3	4	5
4 4 8	7 - 15	9 ◢ 29	4 5 38	6 3 47
6	7	8	9	10
9 ◢ 62	5 2 69	6 1 76	6 3 85	5- 90

《写真8―12》最後の得点が計算され表示

《写真8―13》コメント表示

アナタ ハ 189 テンデ゛ス
****
* GAME OVER *
****
** ナカナカ ヤルネ **

1	2	3	4	5
⧓ 30	⧓ 60	⧓ 89	⧓ 109	9 ◢ 124
6	7	8	9	10
5 - 129	9 ◢ 144	5 4 153	9 ◢ 173	⧓6- 189

モウ イチト゛
シマスカ
Y ? N ?_

《写真8―14》もう一度やる場合はYを入力

行番号40～110：1～9フレームの各第1投, 第2投と10フレームの第1投目

行番号115～190：10フレームの第2投

行番号200～250：10フレームの第3投

行番号300～340：8, 9, 10フレームのスコア表示

行番号360～400：GAME OVER, 評価などの表示

サブルーチン・プログラム

行番号500～520：レーン及びピンの表示

行番号550～635：人間のマークの移動とボールの移動

行番号650～676：ピンをたおす

行番号680～682：ピンをたおす

行番号700～799：スコア表の上段表示

行番号800～826：スコア表の下段（得点）表示

合計5867Byte

《写真8—15》最初の状態にもどる

ゲームの必勝法は別にありません。ヘッドピンをストレートの球でねらうより，フックボールで勝負したほうがよいでしょう。ピンのはねかたはそのつど違います。

ボーリングゲーム・プログラムリスト

```
>L.
   2 C.;P."*** ボ-リング・ゲ-ム ***
";CU.6,4;P."フレ-ム   ボ-ル"
   6 F.Y=3TO5S.2;F.X=4TO18:CU.X
,Y:P.H9A:N.X:N.Y
  10 CU.3,3:P.HAD:CU.12,3:P.HAD
:CU.3,5:P.HAF:CU.12,5:P.HAF:CU.3
,4:P.H8B
  12 CU.5,3:P.HA6:CU.17,3:P.HA6
:CU.5,5:P.HA5:CU.17,5:P.HA5:CU.5
,4:P.H8B
  14 CU.10,3:P.HAE:CU.19,3:P.HA
E:CU.10,5:P.HB0:CU.19,5:P.HB0:CU
.10,4:P.H8B
  16 CU.11,3:P.H20:CU.11,5:P.H2
0:CU.12,4:P.H8B:CU.17,4:P.H8B:CU
.19,4:P.H8B
  18 @(32)=32,@(33)=220,@(34)=2
02,@(35)=204
  20 F.T=0TO31:@(T)=0:N.T:@(46)
=3*9999+2381 :@(47)=@(46)-31,@(4
8)=@(47)-2
  22 @(49)=@(48)-29,@(50)=@(49)
-2,@(51)=@(50)-2,@(52)=@(51)-27,
@(53)=@(52)-2
  24 @(54)=@(53)-2,@(55)=@(54)-
2,@(56)=@(48)+1,@(57)=@(50)+1,@(
58)=@(51)+1
  26 @(59)=@(53)+1,@(60)=@(54)+
1,@(61)=@(55)+1
  30 CU.3,7;P."1   2   3   4
5";CU.3,12;P."6   7   8   9   10
"
  32 F.X=2TO20;F.Y=6TO16S.5;CU.
X,Y;P.H9A;N.Y;F.Y=8TO13S.5;CU.X,
Y;P.H9A;N.Y;N.X
  33 F.Y=7TO15;F.X=1TO21S.4;CU.
X,Y;P.H8B;N.X;N.Y
  34 CU.1,6;P.HAD;CU.21,6;P.HAE
;CU.1,16;P.HAF;CU.21,16;P.HB0;CU
.1,8;P.HA8
  35 CU.1,11;P.HA8;CU.1,13;P.HA
8;CU.21,8;P.HA7;CU.21,11;P.HA7;C
U.21,13;P.HA7
  38 F.X=5TO17S.4;Y=6;CU.X,Y;P.
HA6;Y=16;CU.X,Y;P.HA5;Y=11;CU.X,
Y;P.HA4
  39 F.Y=8TO13S.5;CU.X,Y;P.HA4;
N.Y;N.X
  40 GOS.700;GOS.800;@(0)=@(0)+
1;IF@(0)#10CU.4,4;V=@(0),U=1;GOS
.760;G.60
  50 CU.2,3;P.HAD;CU.2,4;P.H8B;
CU.2,5;P.HAF;CU.3,3;P.H9A;CU.3,5
;P.H9A
  55 CU.3,4;P.H31;CU.4,4;P.H3C
  60 F.T=36TO45;@(T)=1;N.T;CU.1
8,4;P.H31;GOS.650;@(10+@(0))=H;I
F@(0)=10G.115
  70 CU.23,9;IFH=10P.H2A,7D,44,
57,72,78,21,21,2A;G.40
  80 IFH<1P.H2A,76,5E,70,40,70,
21,21,2A;G.90
  85 P.H20,71,44,20,20,20,4B,5F
,5D;CU.27,9;V=10-H,U=1;GOS.760
  90 CU.18,4;P.H32;GOS.650;@(20
+@(0))=H-@(10+@(0));CU.23,9
 100 IFH=10P.H2A,7D,4D,5F,71,21
,21,2A;G.40
 110 P.H2A,2A,50,7D,21,21,2A,2A
;G.40
 115 CU.18,14;IF@(20)=10P.HB7;F
.T=36TO45;@(T)=1;N.T;G.130
 120 IF@(20)=0P.H07;G.130
 125 V=@(20),U=1;GOS.760
 130 CU.18,4;P.H32;GOS.650;CU.1
9,14;IF@(20)=10@(30)=H;G.155
 140 @(30)=H-@(20);IF@(20)=0G.1
65
 155 IF@(30)=10P.HB7;G.200
 160 IF@(20)=10G.170
 165 IF@(20)+@(30)=10P.HB2;G.20
0
 170 IF(@(30)=0)*(@(20)=10)P.H7
0;G.185
 175 IF(@(30)=0)+(@(20)=H)P.H70
;G.185
 180 V=@(30),U=1;GOS.760
 185 IF@(20)+@(30)<10G.300
 190 G.205
 200 F.T=36TO45;@(T)=1;N.T
 205 CU.18,4;P.H33;GOS.650;IF(@
(20)+@(30)=10)+(@(20)+@(30)=20)@
(31)=H;G.225
 220 @(31)=H-@(30)
 225 CU.20,14;IF@(30)=10G.240
 230 IF(@(20)=10)*(@(30)+@(31)=
10)P.HB2;G.300
 240 IF@(31)=10P.HB7;G.300
 245 IF@(31)=0P.H70;G.300
 250 V=@(31),U=1;GOS.760
 300 GOS.800;IF@(19)=10@(9)=@(8
)+10+@(20)+@(30);G.340
 320 IF@(19)+@(29)=10@(9)=@(8)+
10+@(20);G.340
 330 @(9)=@(8)+@(19)+@(29)
```

```
  340 Z=9:GOS.810:@(10)=@(9)+@(2
0)+@(30)+@(31),Z=10:GOS.810
  360 F.X=1TO32:F.Y=1TO5:CU.X,Y:
P.H20:N.Y:N.X:F.Y=6TO16:F.X=22TO
32:CU.X,Y
  370 P.H20:N.X:N.Y:CU.3,2:P."アナ
タ ハ",#1,@(10),"  テンデ゛ス":CU.4,4
  371 IF@(10)=300P."*** パ゛ーフェクト
***":G.378
  372 IF@(10)>=230P."** プ゛ロナ゛ミ テ゛
スネ **":G.378
  373 IF@(10)>=200P."** スバ゛ラシイ テ
゛スネ **":G.378
  374 IF@(10)>=170P."** ナカナカ ヤルネ
 **":G.378
  375 IF@(10)>=140P."** マアマア テ゛ス
ネ **":G.378
  376 IF@(10)>=100P."** ヘタ テ゛スネ
**":G.378
  377 P."モウ ヤメタホウカ゛ イイテ゛スヨ !!"
  378 F.X=20TO32:F.Y=1TO3:CU.X,Y
:P.H2A:N.Y:N.X
  379 CU.21,2:P.H20,07,01,0D,05,
20,0F,16,05,12,20
  380 Y=1,N=0:CU.23,8:P.H53,73,2
0,72,41,44,5E:CU.27,9:P.H7C,4F,7
D,76
  390 CU.24,11:IN."Y ? N "A:IFA=
1G.2
  400 C.:S.
  500 F.Y=1TO15:X=22:CU.X,Y:P.H8
7:X=32:CU.X,Y:P.H84:N.Y
  510 F.T=46TO55:IF@(T-10)=1PO.@
(T),@(35):G.520
  515 PO.@(T),@(32)
  520 N.T:Y=9:F.X=23TO31:CU.X,Y:
P.H20:N.X:R.
  550 B=75,C=@(46)+379,E=C,D=C+1
0
  551 A=P.(7DFCH):IFA=75Y=16:F.X
=23TO31:CU.X,Y:P.H20:N.X:G.555
  553 CU.23,16:P.H0B,20,66,20,75
,7C,45,7B,72:G.551
  555 A=P.(7DFCH):IF(A<73)+(A>76
)A=B
  557 B=A:G.550+(A-72)*10
  560 G.600
  570 PO.E,@(32):IFE>CE=E-1
  575 PO.E,@(33):G.555
  580 PO.E,@(33):G.555
  590 PO.E,@(32):IFE<DE=E+1
  595 PO.E,@(33):G.555
  600 B=75,F=62
  601 A=P.(7DFCH):IF(A<74)+(A>76
)A=B
  602 B=A,@(F)=E,F=F+1:G.600+(A-
73)*10
  610 PO.E,@(32):IFE<@(51)-5R.
  615 E=E-33:PO.E,@(34):G.601
  620 PO.E,@(32):IFE<@(51)-5R.
  625 E=E-32:PO.E,@(34):G.601
  630 PO.E,@(32):IFE<@(51)-5R.
  635 E=E-31:PO.E,@(34):G.601
  650 GOS.500:GOS.550:IF@(74)=@(
46)@(36)=0
  651 IF@(75)=@(47)@(37)=0:G.654
  652 IF@(75)=@(48)@(38)=0:G.654

  653 IF@(75)=@(56)@(36)=0,@(37)
=0,@(38)=0
  654 IF@(76)=@(49)@(39)=0:G.659

  655 IF@(76)=@(51)@(41)=0:G.659

  656 IF@(76)=@(57)@(37)=0,@(39)
=0,@(40)=0:G.659
  657 IF@(76)=@(58)@(38)=0,@(40)
=0,@(41)=0:G.659
  658 IF(@(76)=@(50))*(@(40)=1)@
(40)=0,@(43)=0,@(44)=0,H=39:GOS.
680:H=41:GOS.680
  659 IF(@(77)=@(52))*(@(42)=1)@
(42)=0,H=43:GOS.680:G.670
  660 IF(@(77)=@(55))*(@(45)=1)@
(45)=0,H=44:GOS.680:G.670
  661 IF@(77)=@(59)@(39)=0,@(42)
=0,@(43)=0:G.670
  662 IF@(77)=@(61)@(41)=0,@(44)
=0,@(45)=0:G.670
  663 IF(@(77)=@(53))*(@(42)=1)@(
42)=0,H=42:GOS.680:H=44:GOS.680
:G.670
  664 IF(@(77)=@(54))*(@(44)=1)@(
44)=0,H=43:GOS.680:H=45:GOS.680
:G.670
  665 IF@(77)=@(60)@(40)=0,@(42)
=0,@(44)=0,H=42:GOS.680:H=45:GOS
.680
  670 IF@(75)#@(56)G.675
  671 IF@(73)=@(46)+30@(44)=0,@(
41)=0,H=45:GOS.680:G.675
  672 IF@(73)=@(46)+31@(44)=0,H=
41:GOS.680:G.675
  673 IF@(73)=@(46)+33@(43)=0,H=
39:GOS.680:G.675
  674 IF@(73)=@(46)+34@(39)=0,@(
43)=0,H=42:GOS.680
  675 GOS.500:H=0:F.T=36TO45:IF@(
T)#1H=H+1
  676 N.T:R.
  680 IF@(H)#1R.
  682 @(H)=R.(2):R.
  700 IF@(0)<1R.
  710 Z=@(0):IFZ<6Y=9:G.730
  720 Y=14
  730 IFZ>5Z=Z-5
  740 V=@(10+@(0)),X=2+(Z-1)*4:C
U.X,Y
  750 IFV=0P.H07
  755 U=-1
  760 IFV=1P.H31
  770 IFV=2P.H32
  780 IFV=3P.H33
  785 IFV=4P.H34
  790 IFV=5P.H35
  791 IFV=6P.H36
  792 IFV=7P.H37
  793 IFV=8P.H38
  794 IFV=9P.H39
  795 IFU>0R.
  796 X=X+2:CU.X,Y:IFV=10P.HB7:R

  797 W=@(20+@(0)):IFV+W=10P.HB2
:R.
  798 IFW<1P.H70:R.
  799 V=W,U=1:G.760
  800 IF@(0)<3R.
  801 Z=@(0)-2,S=Z-1:IFS<1S=Z
  802 IF@(10+Z)=10G.805
  803 IF@(10+Z)+@(20+Z)=10@(Z)=@
(S)+10+@(11+Z):G.810
  804 @(Z)=@(S)+@(10+Z)+@(20+Z):
G.810
  805 IF@(11+Z)=10@(Z)=@(S)+20+@
(12+Z):G.810
  806 @(Z)=@(S)+10+@(11+Z)+@(21+
Z)
  810 IFZ<6Y=10:G.814
  812 Y=15
  814 O=@(Z),J=O/100,K=(O-J*100)
/10,L=O-J*100-K*10:IFZ>5Z=Z-5
  818 X=2+(Z-1)*4:CU.X,Y:IFJ=0P.
H20
  820 V=J,U=1:GOS.760:X=X+1:CU.X
,Y:IF(J=0)*(K=0)P.H20:G.824
  822 IFK=0P.H30
  824 V=K:GOS.760:X=X+1:CU.X,Y:I
FL=0P.H30
  826 V=L:GOS.760:R.
>_
```

9

ゲームプログラム入門

ハノイの塔

TK-80BSL I

宮本征治

ハノイの塔というゲームは，もうみなさんよく知っていらっしゃることと思います。

そこで，TK-80BSを使ったBASICでプログラムを組んでみました。

遊び方

まず，プログラムを，カセットテープまたは，キー・ボードより入力します。〔RUN〕と押して復改すると，ゲームが開始します。

まず，**写真9—1**の様に何枚のディスクゲームに挑戦するか聞いてきますので，3～7の数字(3枚～7枚)を入力してください。たとえば5枚と指定すると**写真9—2**の様な表示が出ます。

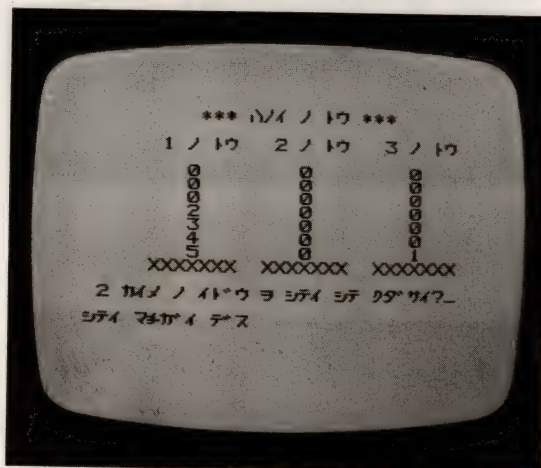
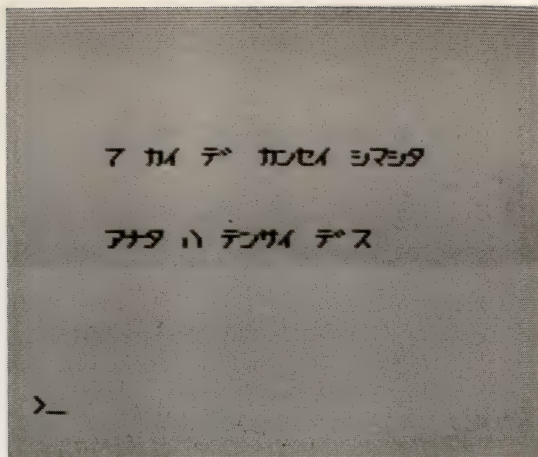
塔の表示には，0とXを使います。ディスクは，1～7の数字で表わしています。数字の大きい方が，大きなディスクを表わします。

ゲームの目的は，1の塔のディスクをそっくり，3の塔に移すことです。ディスクの移動は，2桁の数字で指定します。たとえば写真2のように〔1〕〔2〕と入力すると，1の塔のディスク1が，2の塔へ移動します。

小さなディスクの上に，大きなディスクを乗せることはできませんが，**写真9—3**のように，まちがって小さなディスクに大きなディスクを乗せるような指定をした場合は，プログラムでルール違

*** ハノイ ノ トウ ***
ナンマイ ノ ディスク デ シマスカ?_

《写真9—1》何枚でゲーム

《写真9—2》移動する場所を指定

《写真9—3》移動できない場合は指定し直します

《写真9—4》最小回数で完成した場合のコメント

反をチェックして表示しますので，気にせずゲームを進めてください。

移動が全て完了しますと，**写真9—4**の様に，かかった回数と評価が表示されます。さて，あなたは天才でしょうか，それとも？

プログラムの説明

行番号**10〜125**：初期値の設定。キーボードより入力されるディスクの数Nにしたがって，1の塔(@(1) 〜@(7))にディスクが書き込まれます。Wは移動回数を表わします。

行番号**130〜300**：メインループ。キー・ボードから入力される移動の指定Mにより，DEの値を決定し，ディスク移動処理ルーチン，ディスク処理ルーチンに飛び，ゲームの終了が確認されるまでループを形成します。ゲームの終了は，3の塔の下からN番目の位置にディスクが存在することで確認します。

行番号**500〜540**：塔の表示ルーチン。@(1)〜@(21)を3列に表示するだけです。

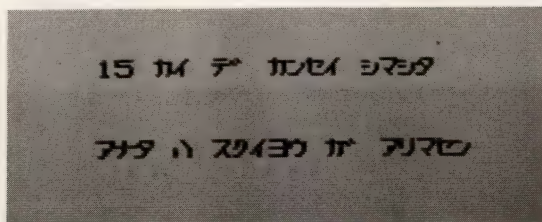
BSの機能をフルに使えば，ディスクの図形を表示することも可能ですが，大小関係が判別しにく

ハノイの塔プログラムリスト

```
   10 C.
   20 CU. 10,1;P."*** ハノイ ノ トウ **
*"
   30 CU. 1,3;IN."ナンマイ ノ ディスク デ
シマスカ"N
   40 IF (N<3)+(N>7) P.;P.;P."デ
ィスク ノ カズ ハ 3マイ カラ 7マイ マデ デス!";
G.30
   50 CU.1,3;P."        1 ノ トウ   2
 ノ トウ    3 ノ トウ"
   60 F.X=1 TO 21
   70 @(X)=0
   80 N.X
   90 F.X=8-N TO 7
  100 @(X)=X+N-7
  110 N.X
  120 GOS.500
  125 W=1
  130 CU.1,14;P.;
  135 CU.1,14
  140 P.#1,W,;IN." カイメ ノ イドウ ヲ
シテイ シテ クダサイ"M
  145 CU.1,16;P."
                   ",
  150 IF M=12 D=1;E=8
  160 IF M=13 D=1;E=15
  170 IF M=21 D=8;E=1
  180 IF M=23 D=8;E=15
  190 IF M=31 D=15;E=1
  200 IF M=32 D=15;E=8
  210 IF D=0 GOS.920;G.130
  220 GOS.800
  250 GOS.500
  260 IF @(22-N)#0 G.1000
  300 G.130
  500 CU.1,5
  510 F.X=1 TO 7
  520 P.@(X),@(X+7),@(X+14)
  530 N.X
  535 P."       XXXXXXX   XXXXXXX
XXXXXXX"
  540 R.
  800 S=0;T=0
  810 F.X=D TO D+6
  815 IF S#0 G.830
  820 IF @(X)#0 S=X;T=@(X)
  830 N.X
  840 IF S=0 G.920
  850 F.X=E TO E+6
  860 IF (@(S)=0)+(@(X)#0) G.900
```

```
  880 IF (@(X+1)>T)+(X=E+6) @(X)
=T;@(S)=0
  900 N.X
  910 IF @(S)=0 W=W+1;R.
  920 CU.1,16
  925 P."シテイ マチガイ デス",
  930 R.
 1000 C.;CU.4,4;P.#1,W-1," カイ デ
カンセイ シマシタ"
 1010 CU.5,8
 1015 P."アナタ ハ ",
 1020 Q=1;F.X=1 TO N
 1030 Q=Q*2
 1040 N.X
 1045 IF W=Q P."テンサイ デス"
 1050 IF(W>Q)*(W<Q+Q/2) P."シュウサイ
デス"
 1060 IF(W>=Q+Q/2)*(W<2*Q) P."ボ
ンサイ デス"
 1070 IF W>=2*Q P."スクイヨウ ガ アリマセ
ン"
```

15 カイ デ カンセイ シマシタ

アナタ ハ スクイヨウ ガ アリマセン

《写真9—5》こんなコメントだと，やる気をなくす

くなることや表示に時間がかかることの二つの理由で，今回は見送りました（読者のみなさんでこのサブルーチンを工夫してみてください)。

800〜930：ディスクの移動処理およびルールチェック・ルーチン。Dはディスクを取る塔，Eはディスクが移動する塔を表わします。これらはメイン・プログラムで，Mの値により初期化されています。Sは移動するディスクの位置，Tはそのディスクの大きさが入ります。

行番号**1000〜1080**：評価プログラム。ハノイの塔は，ディスクの数をnとすると，最小回数は2^n-1であることがわかっていますので，これを基準にして，四段階の評価を行ない，評価のコメントを表示します。

以上でプログラムの説明をおわります。ディスクの数をあまり大きくしすぎて，イライラしないようにしてください。

10 BASICプログラムテクニック

惑星着陸ゲーム

TK-80BSLI

梶原好生

'78の1月の末にTK-80BSを手にして，様々な困難もありましたが何とか私のBSも思いのとおり動いてくれるようになりました。そこでLEVEL-2BASICが走るようになる前のトレーニングとして，いろいろなゲームのプログラムを作っています。ここでご紹介するものは、数週間もかかってあれをやっってみよう，これをつけ足してみようと，追加していったもので，結果的に冗長なプログラムになりました。みなさん方はこれに自由に手を加えて改造してみてください。

このプログラムはCURSOR機能を用い，ダラダラとゲーム進行が画面のスクロールで行なわれるということを避け，実際に宇宙船にのって塔載されたLIBLARY COMPUTERの映し出すMONITORの画面を見ているような気分になれるよう工夫しました。ゲーム自体は割に単純なのですが，ここで用いた手法はBSを使われた方にはグラフィックを用いる上で参考になるのではないかと思っています。いってみればこのプログラムの90%以上はディスプレイのためですから……。

プログラム作成の前に

ここで最初に特殊だと思われるところを先に説明しておきます。BSのLEVEL-1では，X<0, X=0, X>0をＩＦ文で場合けしようとするとうまくいきません。これも暫定版であるからということでがまんして，X>−1000であれば，たとえば次のようにします。

```
IF (X+1000)<1000P. "<" ……………①
IF (X+1000)=1000P. "=" ……………②
IF (X+1000)>1000P. ">" ……………③
```

③は必ずしもこうしなくてよいと思います。もちろんLEVEL-2ではこんなことはしなくてすむと思います。

次にRND関数ですが，私のBSではたとえば，

```
10  FOR T=1 TO 16
20  A=RND (16)
30  PRINT A
40  NEXT T
50  FOR K=1 TO 500;NEXT K
60  GOTO 10
```

としてRND数を調べますと，延々と同じ数が出てくることがあります。このプログラム中ではそれを避けるために配列を用いて前に出たRND数と同じでないかチェックさせています。

これをさせないと，このゲームでは出てくる星がただ１個になってしまって画面がさびしくなってしまいます（じかし，これを入れたため数秒間星が出るまで待つということがちょくちょくあります)。これもインタープリタのステップ数を減らすために仕方なかったということです。

このプログラムを入れたあと，PRINT SIZEにしますと1049程度を表示すると思います。

惑星着陸

それでは着陸の方法の説明です。

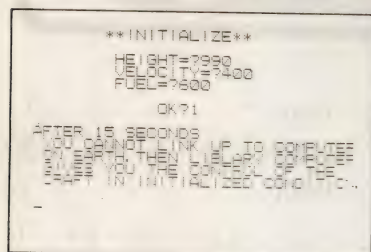
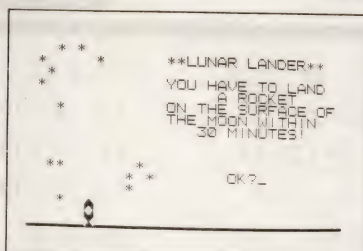
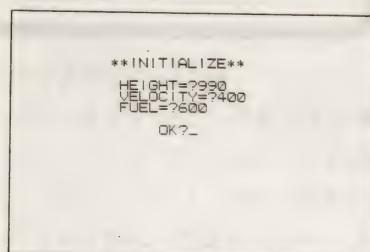
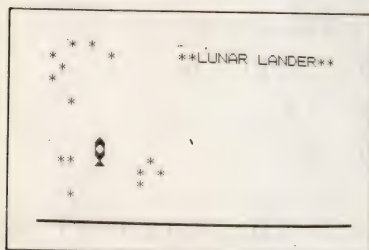
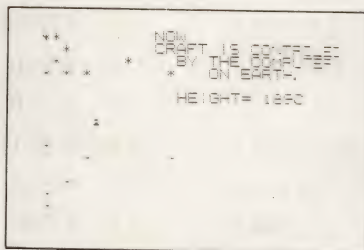
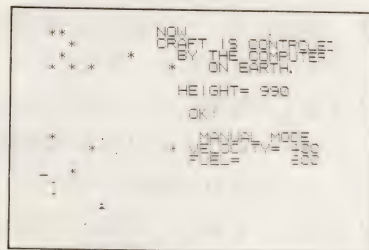
(1)スタートデモンストレーション

プログラムをBSに入れたあと,「R.」とすれば画面はクリアされ，月面着陸船が星空をバックにして月面に向かって降りてきます。**(写真10—A)**。そして月面に着陸したあと画面の右はしに船長に課せられた任務のメッセージが出てきます**(写真10—B)**。30秒以内で月面に着陸せよという任務がわかれば,1を入力していよいよゲームの開始です（以下YES=1，NOは1以外で入力します)。

(2)再び画面がクリアされ，INITIALIZEで初期値を入力します（写真10—C)。

高さは500～1500m，速度は100～1000，燃料は500～1000ぐらいが適当だと思います。

入力した値でよければOKに1を，でなければ1以外を入れれば再び初期値は書きなおせます。OKであればゲーム者（船長）にあたえられた条件が表示されます。**(写真10—D)**。

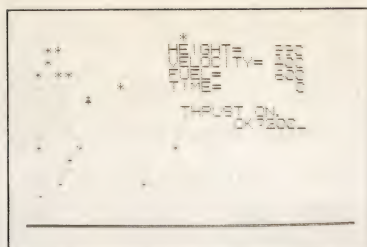
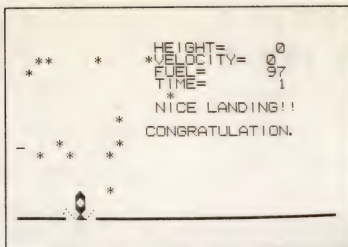
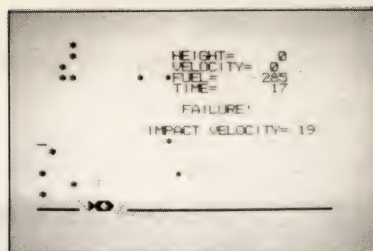
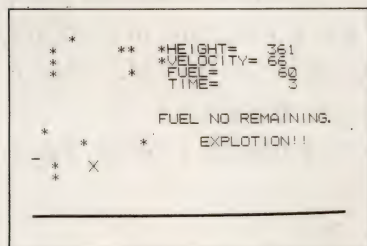
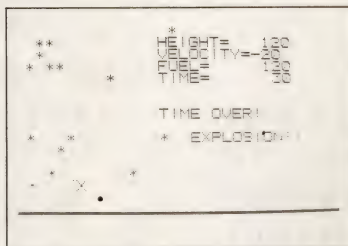
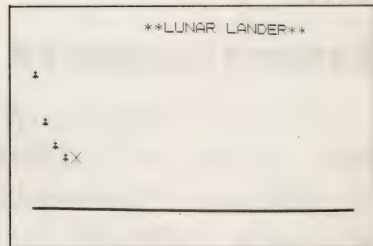
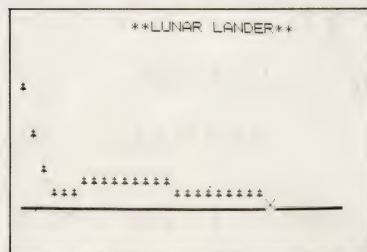
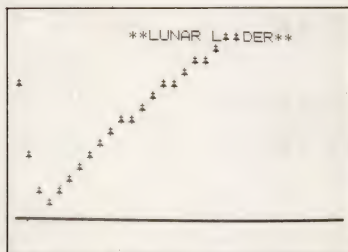
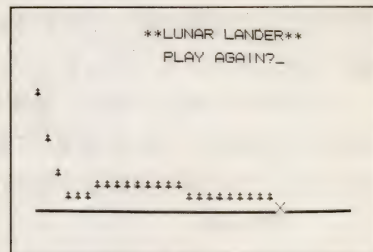
つまり15秒後には地球のコンピュータとの交信が途絶えてそのとき塔載コンピュータは入力された初期値でControlをMANUAL MODEにします。すなわち手動となるのです。

(A)メッセージが送られて来る

(B)着陸ゲームスタート

(C)イニシャライズ。高さと速さを入力

(D) 15秒後地球からのコンピュータ制御が切れる

(E) 自動制御で降下中

(F) マニュアル・モードに切替わったところ

何回ものワープの後，今我々の宇宙船の眼前には大きく赤茶けたβ星の地表がぐんぐんせまりつつある。地球からのコンピュータ制御ももうすぐ切りはなされ，この着陸艇の艇長である私の手にすべてがゆだねられるのだ。

はるか地球ではβ星へのはじめての地球人到達の連絡をかたるをのんで待ちかまえているにちがいない。

ここで着陸に失敗しては今までの努力も水のあわだ。今までのシミュレータによる着陸訓練がモノをいうときだ。地上高“HEIGHT”インプット。速度“VELOCITY”インプット。残り燃料“FUEL”インプット……。ブラウン管にマニュアルモードの表示が出た。

うまく地表で速度0になるよう待っていかなければならない。燃料噴射“THRUST ON OK?”の表示のつど燃料キーを押す。我々の着陸艇はしだいに速度をゆるめながら降下していく。

地表が，小窓いっぱいに拡がってきつつある………。

(G) 逆噴射を入力する。次の時間におけるデータが表示される

(H) 着陸成功！

(I) 着陸失敗。みじめな着陸船の姿が……

(J) 燃料切れで空中分解

(K) 時間切れで空中分解

(L) 初期燃料が足りなかった…

(M) 地表に激突！あわや飛行士は……

(N) ちょっとはでに逆噴射しすぎたかな……

(O) もう一度やりますか？

(3)自動降下

しばらくすると自動的に画面が切り替わり，地球のコンピュータに制御されて星空の中を月面へと降下していきます(**写真10—E**)。ここは腕組みでもして星空をバックに降下していく宇宙船の姿をみていて下さい。なかなかいい感じです。自動降下が終わると(**写真10—F**)となります。

(4)ゲーム・スタート

再び画面が切り替わり，**写真10—G**の月面があらわれていよいよゲーム・スタートです。

逆噴射をTHRUST ON?で聞いてきますので，うまく着陸できるように各秒数ごとに噴射する燃料の量を調整して降下して下さい。VELOCITYがマイナスのときは上向きの速度をもったことになります。100m動くごとに宇宙船は画面上を一コマずつ動きます。着陸に成功すればその勇姿？が**写真10—H**のように現われ，メッセージも表示されます。

失敗すればみじめな姿があらわれ月面衝突速度が表示されます(**写真10—I**)。この速度が少しでも小さければ救いともなります。

次に燃料をなくしたときには，その場で空中分解します(**写真10—J**)。それから30秒(ステップ数)をこえたときも同様に空中分解します(**写真10—K**)(このときは，電源が切れたとでも解釈して下さい)。

30秒以内としたのは，速度がたとえば−100m/sとかなればゲームは終わらなくなってしまうし次の状況表示では横軸に32しかとれないからです。

(5)状況表示

しばらくすると再び画面が切り変わり横軸に時間軸，縦軸に高さをとって今のゲーム状況を表示します。**写真10—L**は初期燃料が少なかったための燃料切れ，**写真10—M**は速度を十分に下げられず地表に激突，**写真10—N**は燃料噴射をやりすぎて逆にまい上がってしまい時間切れでアウト。

このあと画面上でPLAY AGAIN ? と聞きますので1を入力すれば再びゲーム開始です（**写真10—0**）。

プログラムの解剖

以上がゲーム説明ですが，プログラムについて次に説明します。

着陸船の高度および速度の計算

このプログラムのMainをなす部分です。計算を簡単にするために一秒ごとに着陸船をとめ，その時の状態を表示し逆噴射するか否か指令をあおぎます。したがって30秒以内にというのはすなわち30STEP以内で月面に着陸しなさいという意味です。

このような質量の刻々と変わる物体の運動を知るためには次の原理を用いました。

いま着陸船の質量をm(t)，速度をv(t)としてdt時間後に質量dm(<0)が速度v'で物体から離れて飛びだすとこの系の運動量は時間tに

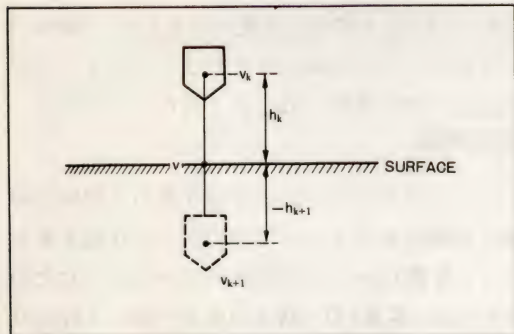
$$Pt=mv$$

t+dtにはこの系の運動量は(m+dm)(v+dv)の他に質量dmがv'の速度をもっていますから

$$Pt+dt=(m+dm)(v+dv)-v'dm$$

運動量の変化がこの系に働く外力の力積Fdtに等しいとおいて，両辺をdtで割ってdt→0の極限をとると

$$\frac{d(mv)}{dt}-\frac{dm}{dt}v'=F$$

したがって

《第10—1図》惑星面についた時の速度

$$\frac{dv}{dt}=\frac{F}{m}+\frac{v'-v}{m}\frac{dm}{dt}$$ となります。

ここで$\frac{dm}{dt}=\mu$とするとこれは，噴射するガスの質量に比例しこれを操作して着陸するわけです。mは時間の関数で$m(t)=mo-\mu t$となりますが，これを用いてvを計算すると

$$v=at-(v'-v)\log\left|1-\frac{\mu}{mo}t\right| \quad a;\ acceleration$$

となりますが，これではTINY BASICでは複雑な計算になってしまいますのでdm<mとしてmを時間的にconstantとしました（LEVEL-2ではこれで計算すると面白いと思います）。

したがって月での重力加速度を$2m/s^2$とすれば

$$\frac{dv}{dt}=2+\frac{v'-v}{m}\mu$$

v'−vのガスのロケットに対する相対速度もロケットの構造で決まる一定値であるとすれば，簡単に次のようになります。

$$\frac{dv}{dt}=2-k\mu \qquad k=構造定数$$

（座標下向きを正とした）

これを用いれば求めようとする高度，速度は次のように決まります。

ある時刻tの速度をvtとすればt+1秒後の速度は

$$v_{t+1}=v_t+2-k\mu$$

次に高度ですがこれは$x=h-v_t-\frac{1}{2}at^2$を用いて，$t=1$，$x=h_{k+1}$，$h=h_k$，$v=v_k$，$a=2-k\mu$であるから

$$h_{k+1}=h_k-\frac{1}{2}(v_t+2-k\mu+v_t)=h_k-1-v_t+\frac{1}{2}k\mu$$

これでもとめる式はすべてまとまりました。

$$a=2-k\mu$$
$$v_{t+1}=v_t+2-k\mu$$
$$h_{k+1}=h_k-1-v_t+\frac{1}{2}k\mu$$

燃料消費の条件をいれると，$k\mu$を消費燃料として

$$\Sigma k\mu \leqq (最初の燃料)$$

で$v=0$，$h=0$とできれば月面着陸に成功したことになります。

塔載する燃料の量には何ら制限はつけませんで

```
>L.
  10 CL.
  20 GOS. 3000
  30 F. T=1 TO 32;W=16
  40 CU. T,W;P. H95;N. T
  50 CU. 15,2;P. "**LUNAR LANDER*
*"
  60 X=7,Y=0
  70 CU. X,Y;P. H20;Y=Y+1
  80 CU. X,Y;P. HBD;Y=Y+1
  90 CU. X,Y;P. HCB;Y=Y+1
 100 CU. X,Y;P. HB8
 110 F. T=1 TO 800;N. T
 115 IF Y=16 G. 130
 120 Y=Y-2;G. 70
>_
```

```
 130 CU. 15,4;P. "YOU HAVE TO LAN
D"
 140 CU. 20,5;P. "A ROCKET"
 150 CU. 15,6;P. "ON THE SURFACE
OF"
 160 CU. 15,7;P. "THE MOON WITHIN
"
 170 CU. 18,8;P. "30 MINUTES!"
 180 CU. 21,12;IN. "OK"X
 190 IF X#1 G. 180
 200 CL.
 210 CU. 8,2;P. "**INITIALIZE**"
 220 CU. 9,4;IN. "HEIGHT="H
 230 CU. 9,5;IN. "VELOCITY="V
 240 CU. 9,6;IN. "FUEL="C
>_
```

```
 250 CU. 13,8;IN. "OK"S
 260 IF S#1 G. 220
 262 CU. 1,10;P. "AFTER 15 SECOND
S"
 264 CU. 2,11;P. "YOU CANNOT LINK
 UP TO COMPUTER"
 266 CU. 2,12;P. "ON EARTH. THEN L
IBLARY COMPUTER"
 268 CU. 2,13;P. "GIVES YOU THE C
ONTROL OF THE"
 270 CU. 2,14;P. "CRAFT IN INITIA
LIZED CONDITION."
 272 F. T=1 TO 9000 ;N. T
 274 CL.
 280 GOS. 3000
>_
```

```
 281 CU. 13,1;P. "NOW"
 282 CU. 13,2;P. "CRAFT IS CONTRO
LED"
 284 CU. 15,3;P. "BY THE COMPUTER
"
 286 CU. 18,4;P. "ON EARTH."
 288 J=H+1500
 290 F=0;Q=7
 300 CU. Q,F;P. H20
 310 F=F+1
 320 CU. Q,F;P. HD7
 330 F. T=1 TO 1300 ;N. T
 334 CU. 15,6;P. #1,"HEIGHT=",J
 336 J=J-100
 340 IF F=16 G. 352
>_
```

```
 350 G. 300
 352 CU. 16,8;P. "OK!"
 354 CU. 17,10;P. "MANUAL MODE"
 356 CU. 16,11;P. #2,"VELOCITY=",
V
 358 CU. 16,12;P. #7,"FUEL=",C
 359 F. T=1 TO 7000 ;N. T
 360 CL.
 370 GOS. 3000
 375 K=0
 380 GOS. 3500
 390 F. G=1 TO 32;CU. G,W
 395 P. H95;N. G
 397 @(49)=1
 400 X=7
>_
```

```
 405 Y=15-H/100
 407 CU. 15,7
 415 @(50+K)=Y
 420 IF Y<=0 G. 440
 422 IF Y>16 G. 460
 425 IF H<=0 Y=16
 430 CU. X,Y;P. HD7
 440 P=@(49+K)
 445 IF P=Y G. 460
 447 IF P>16 G. 460
 450 CU. X,P;P. H20
 460 CU. 16,7;P. "THRUST ON."
 470 CU. 21,8;IN. "OK"F
 480 CU. 20,8;P. " "
 485 Q=H;R=V
>_
```

```
 490 H=H-V-1+F/2;V=V+2-F;C=C-F;
K=K+1
 500 IF K>30 G. 5000
 510 IF(H+5000)=5000 G. 540
 520 IF(H+5000)<5000 G. 4500
 530 IF(C+5000)<=5000 G. 5020
 532 GOS. 3500
 535 G. 400
 540 IF(V+1000)<1000 G. 400
 550 IF(V+1000)>1000 G. 595
 560 IF(C+5000)<5000 G. 5020
 570 G. 4060
 590 G. 400
 595 S=V,H=0,V=0;G. 4520
 600 F. T=1 TO 9000 ;N. T
>_
```

```
 605 CL.
 610 F. T=1 TO 32
 620 CU. T,W;P. H95
 630 N. T
 635 CU. 12,1;P. "**LUNAR LANDER*
*"
 640 F. T=1 TO K+1
 650 L=@(49+T)
 652 IF L=2000 G. 662
 655 IF L=1000 G. 665
 657 IF (L<1)+(L>16) G. 670
 660 CU. T,L;P. HD7;G. 670
 662 Y=16
 665 CU. T,Y;P. HB9
 670 N. T
>_
```

したが，これだけでもなかなかうまく着陸させるのは至難の芸です。こうなればうまくいくという手みたいなものはないように思います。あきらめず挑戦してください。

次に行番号4510について説明します。

これは激突したときの速度をもとめます。ほっておくと高度がマイナスと表示されて激突だということになるのですが，ここではそうはせずに激突だと判定したら高度を0として惑星面についたときの速度を補間法によってもとめ，その速度を表示します。

第10―1図により惑星面についた時の速度は

```
-.330
 330 F.T=1 TO 6000 :N.T
 -30 CU.14,3:IN."PLAY AGAIN"S
 710 IF S=2 S.
 720 IF S=1 G.200
 730 IF (S#1)+(S#2) G.700
3000 F.I=2 TO 31
3010 @(I)=R.(16)
3015 IF @(I)=7 G.3010
3020 IF @(I)=@(I-1) G.3010
3030 N.I
3040 F.I=1 TO 15
3050 CU.@(I+1),@(I+16):P.H2A
3060 N.I:RET.
3500 CU.15,2:P.#5,"HEIGHT=",H
```

```
3510 CU.15,3:P.#3,"VELOCITY=",
3520 CU.15,4:P.#7,"F.E.",C
3525 CU.15,5:P.#7,"TIME=",
3550 RET.
4000 CU.20,9:P."WAIT"
4003 CU.16,7:P."  "
4005 F.T=1 TO 2000 :N.T:CL.
4010 GOS.3000
4030 F.T=1 TO 32
4030 CU.T,W:P.H95:N.T
4040 GOS.3500
4050 RET.
4060 CU.X,Y:P.H20:CU.7,16:P.HDT
_
```

```
4065 GOS.4000
4070 CU.7,16:P.HB8:CU.6,16:P.HF
E
4080 CU.7,15:P.HCB:CU.8,16:P.HF
-
4090 CU.7,14:P.HBD
4100 CU.15,7:P."NICE LANDING!!"
4110 CU.14,9:P."CONGRATULATION.
4115 @(50+K)=16
4120 G.600
4130 CU.8,16:P.HBA:CU.9,16:P.HF
-
4140 CU.17,8:P."IMPACT!!"
_
```

```
4150 RET.
4500 CU.X,Y:P.H20:CU.7,16:P.HB9
4510 S=(Q*V-H*R)/(Q-H):H=0:V=0
4515 @(50+K)=2000
4520 GOS.4000
4530 CU.6,16:P.HB7:CU.5,16:P.HF
E
4540 CU.7,16:P.HCB:CU.7,14:P.HF
B
4550 CU.8,16:P.HBA:CU.9,16:P.HF
D
4560 CU.16,7:P."FAILURE!"
4570 CU.12,9:P.#1,"IMPACT VELOC
ITY=",S
>_
```

```
4575 @(50+K)=2000
4580 G.600
4590 CU.Y,X:P.HB9
5000 CU.15,8:P."TIME OVER!"
5003 CU.15,7:P."  "
5005 CU.18,10:P."EXPLOSION!!"
5007 @(50+K)=1000
5009 CU.X,Y:P.HB9
5010 G.600
5020 CU.14,8:P."FUEL NO REMAINI
NG."
5022 CU.15,7:P."  "
5025 CU.X,Y:P.HB9
5030 CU.18,10:P."EXPLOTION!!"
5032 @(50+K)=1000
>_
```

```
5040 G.600
>_
```

$$v=\frac{v_k v_{k+1}+(-h_{k+1})v_k}{h_k+(-h_{k+1})}$$

したがってプログラム中では

$$V=\frac{Q*V*-H*R}{Q-H}$$

となります（あまり大きな降下速度で衝突するとこの式でエンザンエラーを起こしてしまいます）。

ディスプレイ部について少し説明します。

行番号3000～3060は星を表示するためのルーチンです。何もこんなことはやらなくてもよいのですが，3015みたいな苦しいことはやりたくないという向きは，一回着陸船を動かすたびに3040とよべば着陸船と重なってしまったために消えてしまった星は更新されて再びあらわれます。そのためあえてこのような形にしました（このときはスタートデモンストレーションのところを少しかえなければなりません。）

成功，失敗のとき宇宙船の大きさが前と変わります（要するに拡大率が変わる）ので背景の星が同じままではおかしいと考えて，この時は画面をクリアして新しく書きかえていますが，時間かかるので各自の好みでかえて欲しいところです。

プログラム作りのスリル

プログラムの説明は以上で終わりますが、作っていて感じたことを2，3書いておきます。同じBASICをもっている人，これから持つ人にも参考になったらと思います。

(1) グラフィックのときはPRINT“　”という形ではなく，Hコードでキャラクターを指定していった方が無難です（P.“　”ではその後の文字を消してしまう）。

(2) 複雑なプログラムを作るときは，アルファベットの表をつくってどの文字をどこでどういう変数に使ったかCHECKしておくとまちがいが

なくなると思います。

(3) グラフィックで画面を構成するときはレイアウトしてからにすれば結局時間の節約になります。

(4) H20で指定する文字すなわちblank(これは何もないというひとつの特殊記号）を効果的に使うとよいと思います。(1)とのかねあいでこれを使うことは重要なテクニックです。

以上ですがLEVEL-1程度だとこのようなグラフィックを用いたゲームを作るにはとても都合よいと思います。

このプログラムに関しては，かなりひとりよがりで作りましたのでTRACEするのは骨が折れるかもしれませんが，思わぬところに虫がいるかも知れませんので，そこは実際にやられる方にまかせておきたいと思います。

今私はプログラムを作ることが最大のスリリングなゲームではないかという気がしています。何回やってもプログラムが通らない。何故だろうと考えるその醍醐味は“惑星着陸ゲーム”の比ではありません。

ですからできたゲームで遊ぼうという気はできてしまえばうすれてしまいます。今は友達を連れこんで，どうだ，面白いだろうとおしつけることで，“間接的？”にこのゲームを楽しんでおります。

11 画面効果を活かした

モグラたたきゲーム

TK-80BSL II

吉沢　優

遊び方

このモグラタタキ・ゲームは，いくつかの穴から出てくるかわいい（？）モグラを，その穴を入力することによって退治するゲームです。それでは，もっと詳しく説明しましょう。

まず，プログラムを入力したのち〔RUN〕を入力すると，下のように尋ねてきます。そこで自分の技量にあわせた数値を入力するわけですが，この数値は，モグラの出ている（現われている）時間を表すものなので，1～5ぐらいが適当です。数が小さければ出ている時間も短く，大きければ長くなります。

復改を入力すれば，すぐゲーム開始になります。画面には10個の穴が表示され，その穴の1か所より，モグラが出てきますので，その穴の番号を出ている間に入力すればいいのです。

もし当っていれば，モグラがつぶれた表示を，

もし当っていなければモグラが，「あっかんべー」をしてそれぞれ結果を表示します。

ナンコースニ シマスカ?_

《写真11―1》コース入力

ゲーム数は20ゲームですが，つぶれたモグラの数が15以上にならなければ，再ゲームはできません。

画面，右上のLEVELとは，はじめに入力したコース数です。そして一番下の行のSCOREは，当たった数，LEAVEは残りのゲーム数です。さてあなたのスコアーは？

プログラム説明

使用した文字変数は，以下の通りです。

A：キーボードより入力される値
B：ゲーム数（初期値）
C：コース
D：モグラの位置（RND）
E：タイマー

《写真11―2》当り

《写真11―4》はずれ

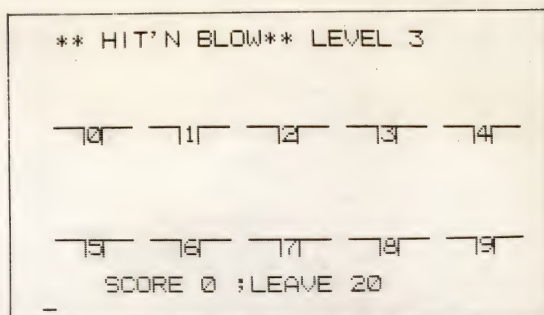
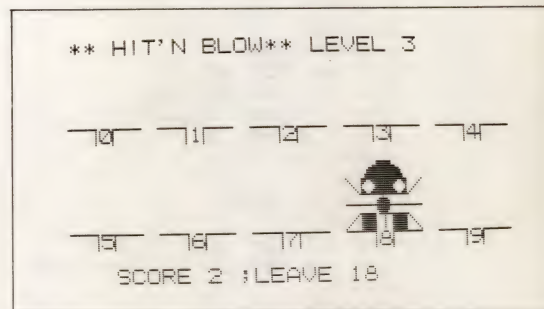
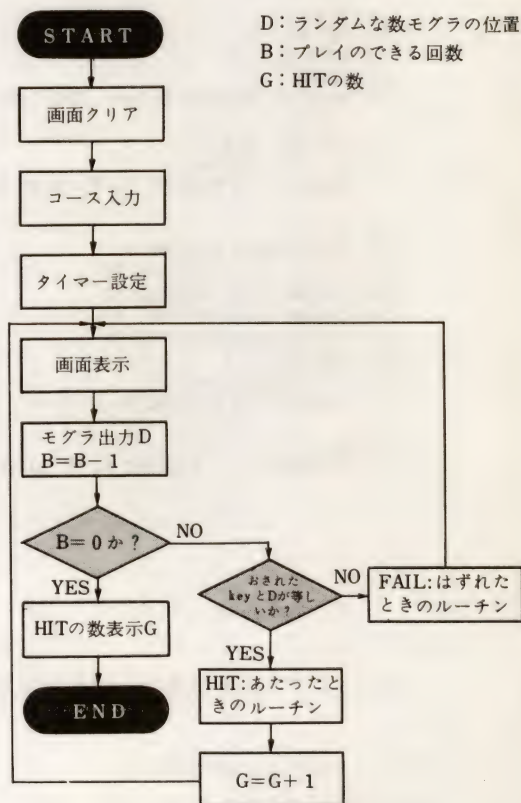
F：Aを0～9に変換した値
G：HIT数
IP：タイマー
M：コース相応タイマー
U：再ゲーム入力用
K：画面作成に使用
行番号30～40：ポートA（キーボード）からの入力を読む。
行番号50～70：結果に相応したルーチンへジャンプ。
行番号650～660：モグラの位置を決める乱数DによってX，Y座標を決める。
行番号1000～1070：画面表示ルーチン。
行番号4000～4300：あたったときの表示ルーチン。
行番号5000～5200：はずれたときのルーチン。
行番号6000～END：エンド・メッセージなど。

プログラムを作ってみて

やはり経験不足なのか，プログラムの製作でもかなり苦労しました。たとえば，キーボードから

《第11―1表》機械語で乱数を作る

アドレス	マシン語
8200	：21，0F，82，7E，87，87
06	：4F，86，77，79，87，86
0C	：3C，77，C9

```
11  CALL  8200H
12  S=PEEK(820FH)
13  D=S*10/257
```

《写真11―4》10個のモグラの穴が

《写真11―5》モグラがピョコン

《第11―1図》モグラたたきのフローチャート

```
  3 LET B=20
  4 LET G=0
  5 CLEAR : INPUT "ﾅﾝｺｰｽﾆ ｼﾏｽｶ"C
  8 LET M=C*100
 10 GOSUB 995
 11 RANDOMIZE
 12 LET D1=RND(10)-1
 13 LET D=INT(D1)
 15 FOR P=1 TO M
 16 NEXT P
 17 GOTO 600
 30 LET A=PEEK(7DFCH)
 35 IF A<48 THEN  GOTO 36
 36 IF A>57 THEN  GOTO 38
 37 GOTO 40
 38 LET A=F
 40 LET F=(A-48)
 50 IF B=0 THEN  GOTO 6000
 55 IF D=F THEN  GOTO 4000
 60 IF F<>D THEN  GOTO 5000
 65 IF F<>A THEN  GOTO 5000
 70 GOTO 5000
600 REM
610 FOR O=1 TO 1000
620 NEXT O
650 IF D<5 THEN  LET X=D*6+2,Y=2
660 IF D>4 THEN  LET X=(D-5)*6+2,Y=8
700 REM
705 CURSOR X,Y
710 PICTURE 20,C2,80,C3,20
715 LET Y=Y+1: CURSOR X,Y
720 PICTURE B6,CB,80,CB,B3
725 LET Y=Y+1: CURSOR X,Y
730 PICTURE 9A,9A,DE,9A,9A
735 LET Y=Y+1: CURSOR X,Y
740 PICTURE B3,80,14,80,B6
750 FOR I=1 TO M
760 NEXT I
763 LET B=B-1
765 GOTO 30
995 CLEAR
1000 PRINT " ** HIT'N BLOW** LEVEL",#1,
C
1010 FOR K=0 TO 4
1015 LET X=K*6+2,Y=6
1020 CURSOR X,Y: PICTURE 96,9F,20,9E,96

1025 NEXT K
1035 CURSOR 4,6: PICTURE 30
1036 CURSOR 10,6: PICTURE 31
1037 CURSOR 16,6: PICTURE 32
1038 CURSOR 22,6: PICTURE 33
1039 CURSOR 28,6: PICTURE 34
1040 FOR K=0 TO 4
1045 LET X=K*6+2,Y=12
1050 CURSOR X,Y: PICTURE 96,9F,20,9E,96

1055 NEXT K
1060 CURSOR 26,12: PICTURE 96,9F,20,9E,
96
1064 CURSOR 4,12: PICTURE 35
1065 CURSOR 10,12: PICTURE 36
1066 CURSOR 16,12: PICTURE 37
1067 CURSOR 22,12: PICTURE 38
1068 CURSOR 28,12: PICTURE 39
1070 CURSOR 5,14: PRINT "SCORE",#1,G,#2
,":",#2,"LEAVE",#1,B
1080 RETURN
4000 CLEAR : CURSOR 11,1: PICTURE 8B,20
,8B,8B,20,96,14,96
4010 CURSOR 11,2: PICTURE A8,9A,A7,8B,2
0,20,8C,20
4020 CURSOR 11,3: PICTURE 8B,20,8B,8B,2
0,20,8C
4030 CURSOR 15,6: PICTURE 2A: CURSOR 13
,7: PICTURE 2A,20,8B,20,2A
4040 CURSOR 12,8: PICTURE 2A,20,B6,20,B
3,20,2A
4050 CURSOR 13,9: PICTURE B6,20,20,20,B
3
4060 CURSOR 12,10: PICTURE 90,9D,C2,80,
C3,9D,9D
4100 FOR E=1 TO 1000
4150 NEXT E
4200 LET G=G+1
4300 GOTO 10
4400 LET C=0: GOTO 8
5000 CLEAR : CURSOR 10,1: PICTURE AD,9A
,AE,20,C1,20,20,A6,20,88
5010 CURSOR 10,2: PICTURE A8,9A,20,B3,2
0,B6,20,8B,20,88
5020 CURSOR 10,3: PICTURE 8B,20,20,9E,9
6,9F,20,A5,20,A0,9D
5030 CURSOR 13,6: PICTURE C2,80,C3
5035 CURSOR 12,7: PICTURE B6,CB,20,30,B
3
5040 CURSOR 12,8: PICTURE 9A,9A,CA,B6,B
6,C1
5050 CURSOR 12,9: PICTURE B3,A6,A6,B3,2
0,8F
5060 CURSOR 13,10: PICTURE 8B,8B,88,20,
20,B6
5070 CURSOR 13,11: PICTURE C8,9B,B6,9D,
20,20,BE
5080 CURSOR 17,12: PICTURE B6,B3
5100 FOR E=1 TO 1000
5150 NEXT E
5200 GOTO 10
6000 CLEAR : CURSOR 8,3: PRINT "***GAME
OVER***"
6010 CURSOR 10,5: PRINT "ｼﾝﾀﾞ ﾓｸﾞﾗ",#1,
G
6020 IF G>15 THEN  CURSOR 10,6: PRINT "
VERY GOOD! YOU ARE THE CHAMPION"
6030 IF (G<15)*(G>=10) THEN  CURSOR 10,
6: PRINT "ﾏｱﾏｱ ﾀﾞﾖ!"
6040 IF G<10 THEN  CURSOR 10,6: PRINT "
  "
6050 IF G>=10 THEN  CURSOR 5,8: INPUT "
===PLAY AGEIN(Y OR N)==="T
6060 IF T=Y THEN  GOTO 3
```

の入力とか，二つの条件を満たす命令などです。これらは自分なりにいじってみたり，Bit-INNにいったり，他のプログラムを参考にしたりして解決しました。

苦労して気づいた点は，画面構成だとか，グラフィック記号を使用する絵を作るときに,「グラフィック・パッド」というのを使うということです。これは文房具店にあります。

このプログラムに限りませんが，改良することによって，より楽しいものがいくらでもできると思います。マイコンはユーザーにいくらでも対応してくれるのだから，ぜひ一度，自分でプログラムを組み，マイコンを「自分のもの」という意味のマイコンにしていくといいと思います。

12

効果的キー入力活用

サブマリンゲーム

TK-80BSL I

野口武志

はじめに

よく町のゲームコーナーでみかけるのに潜水艦ゲームがあります。潜望鏡をのぞいて，スコープ内を走り回わる駆逐艦や魚雷船を命中させるゲームです。このプログラムは，反対に駆逐艦から爆雷を投下して，海中にいる潜水艦を破壊するゲームです。

駆逐艦の移動や爆雷の投下はすべてBSのフルキー・ボードから入力します。効果音のプログラムもありますのでゲームのスリルは最高です。

ゲーム開始

まず，プログラムを入力してRUNさせますと，画面にタイトルが表示されて「S」を押してくださいと文字が出ます。

Startの「S」を押すことによってゲーム開始です。キー・ボード（1~5）と画面ディスプレイの関係は，

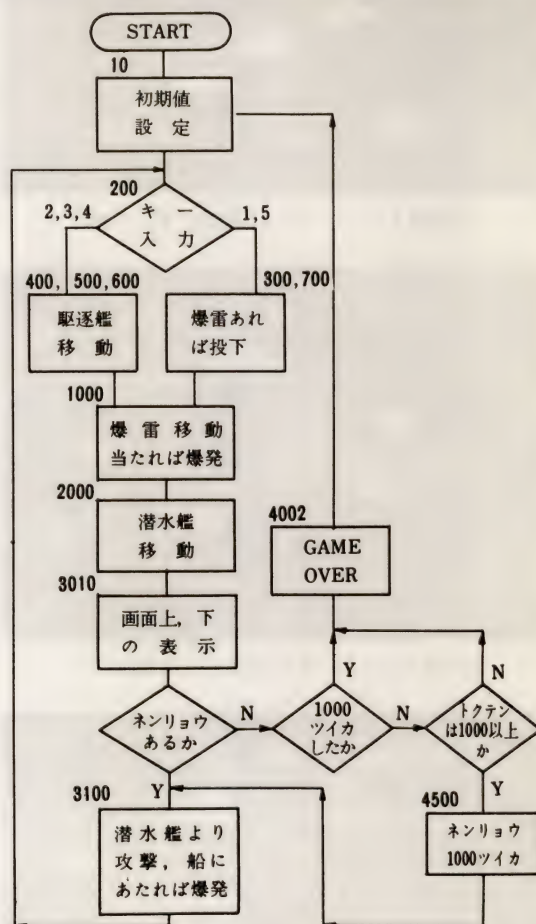
1：爆雷を駆逐艦の左より投下

2：駆逐艦を左へ移動

3：駆逐艦を停止

4：駆逐艦を右へ移動

5：爆雷を駆逐艦の右より投下

となっています。それぞれ1~5のキーを押して攻撃のしやすい所へ駆逐艦を移動させてください

《第12—1図》サブマリンゲーム・フロー

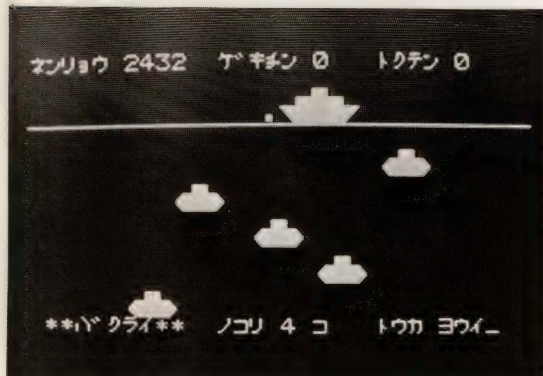
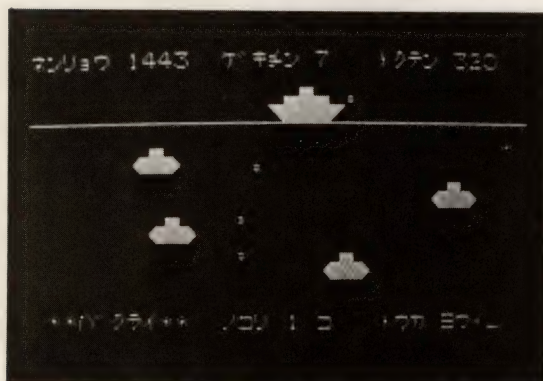
（一度爆雷を投下すると2, 3, 4のどれかのキーを押していないと爆雷は投下できません）。「トウカOK！」の表示の時，爆雷は投下できます。

ゲームのルール

駆逐艦の持っているネンリョウは3,000からスタートしていきます。ネンリョウがなくなったら，その時点で“GAME OVER”となります。ただし，その時点で得点が1,000以上あれば，ネンリョウが1,000新たに追加されます。

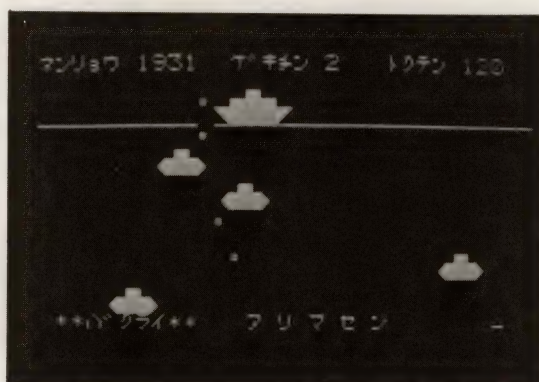
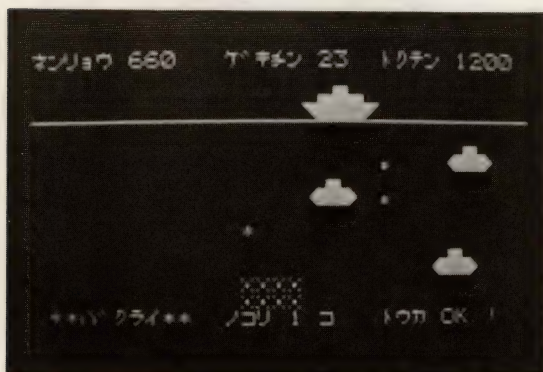
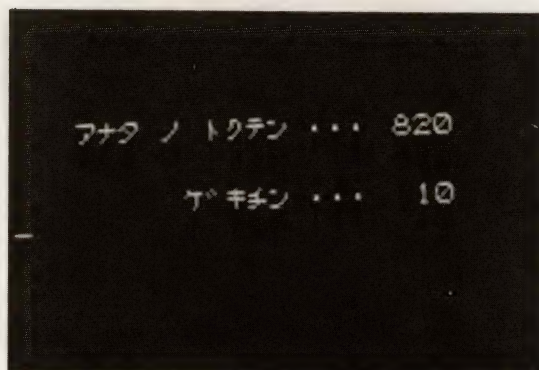
海中に現われる潜水艦は最高5隻です。潜水艦のスピードは，海面から下にいくにつれ遅くなります。得点は海面に近い順に20点，40点，60点，80点，100点となっています。一番海底に近い潜水艦に当たれば得点が多く加算されます。

潜水艦もただ現われて海底を移動してはいません。時々潜水艦から駆逐艦に対してミサイル「＊」が発射されます。もしこの攻撃を受けると，今まで撃沈した数×20点が得点よりひかれます。さらに撃沈した潜水艦の数も半分となってしまいます。

ネンリョウがなくなって“GAME OVER”になると，得点した数×20がプラスされ，それが最終

《写真12—1》ゲーム開始「S」を押す

《写真12—2》爆雷を投下（船尾より）

《写真12—3》みごと命中

《写真12—4》潜水艦からのミサイル攻撃

《写真12—5》駆逐艦総攻撃

《写真12—6》ゲーム終了

得点となります。

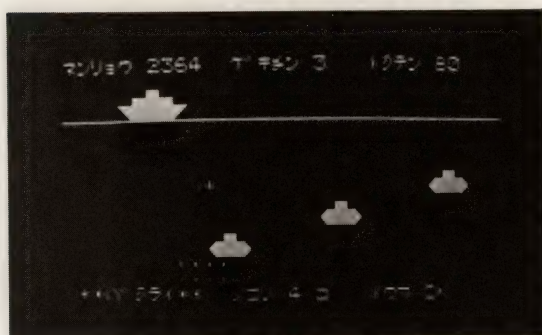
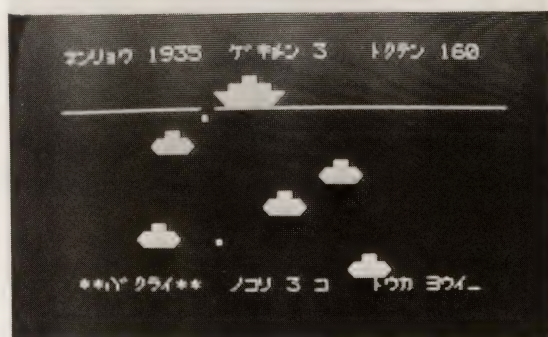
画面の下の＊＊バクライ＊＊の後の数字はバクライの状態をしめます。バクライを投下するごとにマイナス1されます。最高5個のバクライが投下できます（ただし爆発したり，海底までいったら爆雷の数は加算されます。）

“トウカヨウイ”の表示の時は「1」や「5」を押しても爆雷は投下されません。その場合は，まず「2」「3」「4」のキーどれかを押して“トウカOK！”と表示されてから，「1」「5」のキーを押して爆雷を投下してください。

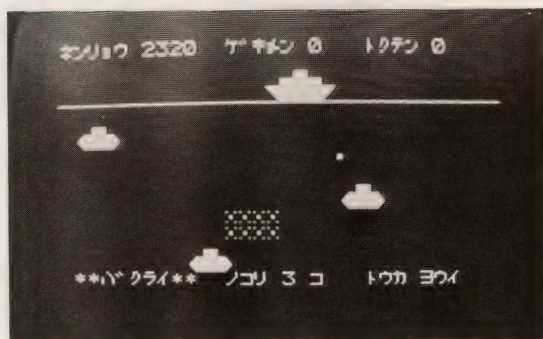
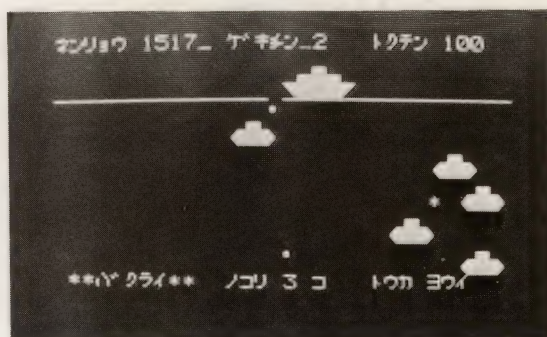
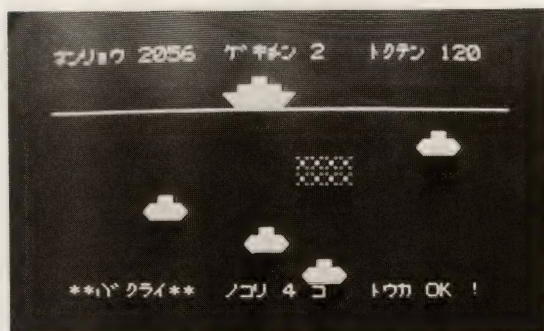
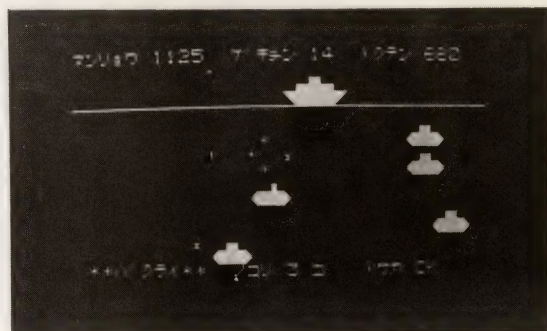
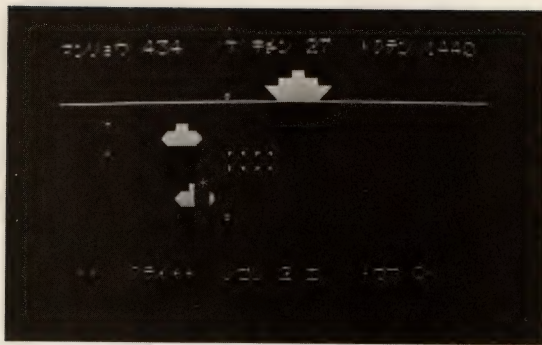
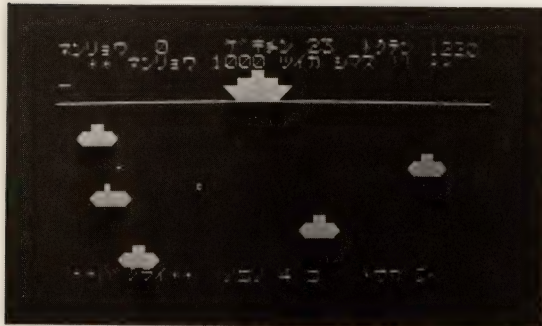
またこのゲームでは音出しのサブルーチンもあり，音を出してゲーム効果を上げることができます。

音が発生するのは，爆雷の投下のとき，潜水艦撃沈の時，駆逐艦撃沈の時，ネンリョウを新たに1,000追加する時にでます。

この音発生のサブルーチンは機械語によって8200番地より8235番地に書かれています。オーディオアンプはPPIのポートC，ビット1にコンデンサ（0.01μF）を入れてつないでください（くわしくはTK-80E/80応用プログラムP.13）。

ゲーム終了

《写真12—7》マイコン大海戦スタート

《写真12—8》STEP1

《写真12—9》STEP2

《写真12—10》STEP3

《写真12—11》STEP4

《写真12—12》STEP5

《写真12―13》STEP6

《写真12―14》トクテン表示

ネンリョウが0になるとゲームは終了です。ゲームの得点がもし最高点だったら，今までの最高点といれかわります。もちろん，プログラムを入力して最初にゲームをした人が最高点ですので，最初はその人の得点が最高点としてマイコンに登録されます。次からゲームをする人は，その最高を目指して挑戦してください。得点の表示は撃沈した順に加算されて表示しますので，合計得点が表示されるまでスリル満点です。

《写真12―15》最高得点獲得

ゲームが終了すると「S」を押してくださいと表示がでますので，何人でもゲームを楽しむことができます。

色つきのセロファンを買って帯状にして潜水艦の移動する所にはっておくと，色によって得点の差が見やすくなります。一番底の潜水艦を青にしたり，中頃を黄にしたりすると，ねらいもつけやすいでしょう。「命中，イエローサブマリン」というところです。爆雷も深く落ちるにつれ色が変化してゲームのおもしろさもふえるでしょう。

```
   1 G.4
   2 CU.X,Y;P.H20,20,20,20;Y=Y-
1;CU.X,Y;P.H20,20,20,20;R.
   3 CU.X,Y;P.H20,BC,80,80,BA,2
0;Y=Y-1;CU.X,Y;P.H20,20,A9,AA,20
;R.
   4 @(22)=0;C.
   5 F.X=5TO26;F.Y=3TO7;CU.X,Y;
P.H2A;N.Y;N.X;F.X=6TO25;F.Y=4TO6
;CU.X,Y;P.H20
   6 N.Y;N.X;CU.8,5;P."センスイカン ケ
゛キチン ゲーム",
  10 A=P.(7DFCH);IFA#83CU.3,16;
P."スタート ボタン (S) ヲ オシテクダサイ !";
G.10
  15 C.
  20 Y=5;F.X=1TO32;CU.X,Y;P.H96
;N.X
  30 Z=15,U=0
  32 F.T=1TO10;@(T)=0;N.T;@(20)
=0,@(21)=0
  34 I=-R.(10),K=28+R.(10),M=-R
.(5),O=28+R.(5),Q=-R.(3)
  35 B=51,C=3000 ,E=1
  36 @(13)=32,@(14)=42,@(15)=15
0,@(16)=97,@(0)=6*5000+2223
  45 CU.1,1
  50 P."ネンリョウ          ゲキチン      ト
クテン"
  60 CU.2,16;P."**ハジメ**",
 200 A=P.(7DFCH);IF(A<49)+(A>53
)A=B
 210 B=A;G.(A-46)*100
 300 W=@(0)+96+Z;G.710
 400 IFZ>1Z=Z-1
 500 S=1,Y=4;CU.Z,Y;P.H20,85,80
,80,80,81,20;Y=3;CU.Z,Y;P.H20,20
,91,80,91,20,20
 510 G.1000
 600 IFZ<26Z=Z+1
 610 G.500
 700 W=@(0)+102+Z
 710 IFS=0G.1000
 720 S=0,V=1
 730 IF@(V)=0@(V)=W;G.800
 740 V=V+1;IFV=6G.1000
 750 G.730
 800 CA.8206H
1000 F.V=1TO5;IF@(V)=0G.1200
1010 Y=@(V)-32;IF(Y>@(0)+160)*(
Y<@(0)+193)PO.Y,@(15);G.1020
1015 PO.Y,@(13)
1020 X=P.(@(V));IF(X=128)+(X=18
6)+(X=188)+(X=169)+(X=170)G.1050

1040 PO.@(V),@(16);G.1180
1050 IF@(V)<@(0)+257X=I+1,Y=6,I
=-R.(10),@(20)=@(20)+20;G.1100
1060 IF@(V)<@(0)+321X=K+1,Y=8,K
=28+R.(10),@(20)=@(20)+40;G.1100

1070 IF@(V)<@(0)+385X=M+1,Y=10,
M=-R.(5),@(20)=@(20)+60;G.1100
1080 IF@(V)<@(0)+449X=O+1,Y=12,
O=28+R.(5),@(20)=@(20)+80;G.1100

1090 X=Q+1,Y=14,Q=-R.(3),@(20)=
@(20)+100
1100 CU.X,Y;P.HFA,FA,FA,FA;Y=Y+
1;CU.X,Y;P.HFA,FA,FA,FA
1110 @(21)=@(21)+1;F.T=1TO5
1120 CA.8200H
1130 CA.8206H
1140 N.T
1150 GOS.2;@(V)=0;G.1200
1180 IF@(V)>@(0)+480PO.@(V),@(1
3);@(V)=0;G.1200
1190 @(V)=@(V)+32
1200 N.V
2000 Y=7;IFI>27X=I+1;GOS.2;I=-R
.(10);G.2200
2010 I=I+1;IFI<1G.2200
2020 X=I;GOS.3
2200 Y=9;IFK<2X=K+1;GOS.2;K=28+
R.(10);G.2400
2210 K=K-1;IFK>28G.2400
2220 X=K;GOS.3
2400 IFU=1G.2805
2405 Y=11;IFM>27X=M+1;GOS.2;M=-
R.(5);G.2600
2410 M=M+1;IFM<1G.2600
2420 X=M;GOS.3
2600 Y=13;IFO<2X=O+1;GOS.2;O=28
+R.(5);G.2800
2610 O=O-1;IFO>28G.2800
2620 X=O;GOS.3
2800 G.3000
2805 Y=15;IFQ>27X=Q+1;GOS.2;Q=-
R.(3);G.3000
2810 Q=Q+1;IFQ<1G.3000
2820 X=Q;GOS.3
3000 U=U+1;IFU=3U=0
3010 CU.6,1;P.#1,C,
3020 C=C-R.(50);IFC<0CU.7,1;P."
0",;G.4000
3025 D=0
3030 F.V=1TO5;IF@(V)=0D=D+1
3040 N.V
3050 CU.18,1;P.#1,@(21);
3060 CU.27,1;P.#1,@(20);
3065 CU.13,16
3070 IFD=0P." アリマセン           "
,;G.3100
3080 P."ノコリ",#1,D," コ",;CU.22,1
6;IFS=1P." トウカ OK !",;G.3100
3090 P." トウカ ヨウイ",
3100 V=6
3105 IF@(V)#0G.3130
3110 Y=@(0)+192+R.(320);X=P.(Y)

3120 IF(X#32)*(X#97)@(V)=Y;G.31
40
3130 V=V+1;IFV#11G.3105
3140 F.V=6TO10;IF@(V)=0G.3300
3150 X=@(V),@(V)=@(V)-32;IF(X>@
(0)+160)*(X<@(0)+193)PO.X,@(15);
G.3200
3160 PO.X,@(13);PO.@(V),@(14);G
.3300
3200 X=P.(@(V));@(V)=0;IF(X=32)
+(X=97)G.3300
3210 X=Z+1;Y=4;CU.X,Y;P.HFA,FA,
FA,FA,FA;Y=3;CU.X,Y;P.HFA,FA,FA,
FA,FA
3220 @(20)=@(20)-@(21)*20,@(21)
=@(21)/2
3230 F.T=1TO7
3235 CA.8206H
3236 CA.8206H
3240 CA.8200H
3250 N.T
3300 N.V;G.200
4000 IF(E=1)*(@(20)>=1000)E=0,C
=1000 ;G.4500
4002 C.
4010 CU.4,9;P."アナタ ノ トクテン ...",
#1,@(20)
4012 CU.9,12;P."ゲキチン ...",#3,@
(21)
4013 CA.8206H
4014 IF@(21)#0@(21)=@(21)-1,@(2
0)=@(20)+20;F.T=1TO1000 ;N.T;G.4
010
4016 IF@(20)>@(22)@(22)=@(20);C
U.15,9;P."ハ サイコウ デス !!"
4018 CU.5,12;P."サイコウ トクテン ...",
#1,@(22)
4020 F.X=5TO27;F.Y=1TO5;CU.X,Y;
P.H2A;N.Y;N.X
4030 F.X=6TO26;F.Y=2TO4;CU.X,Y;
P.H20;N.Y;N.X
4040 CU.8,3;P."G A M E   O V E
R",
4100 G.10
4500 F.T=1TO10
4520 CU.3,2;P."** ネンリョウ 1000 ツイ
カ シマス !! **"
4530 CA.8206H
4540 CA.8200H
4550 CU.3,2;P.;
4560 N.T;G.3025
```

*DM, 8200, 8235

8200	01	10	20	C3	09	82	01	05
8208	10	F5	C5	D5	E5	2E	FF	61
8210	48	3E	02	D3	FA	2B	7C	A7
8218	CA	31	82	0D	C2	15	82	48
8220	3E	00	D3	FA	2B	7C	A7	CA
8228	31	82	0D	C2	24	82	C3	10
8230	82	E1	D1	C1	F1	C9		

13

機械語による応用

HIT & BLOW

EX-80機械語

佐藤　隆

数当てゲームの古典として有名なヒットエンドブローのプログラムです。

HIT & BLOW とは

HIT & BLOWとはMASTER MINDともいって数当てゲームのことです。

MASTER (EX-80) が4桁の数 (MIND) をランダムに発生し，それをプレーヤがいかにはやく当てるかを競うゲームです。

4桁の数には同じ数字は2個含まれていません。

すなわち，4桁とも異なる数字の組合せになっています。プレーヤが数 (MIND) を当てる手助けとしてMASTER (EX-80) はHITの数とBLOWの数を表示してきます。“HITとは数字もその数字の位置も合っている数字の数のこと”です。

また“BLOWとは数字の位置は違っているが，正解の中に含まれている数字と同じ数字が回答の数字の中に含まれているときの数”です。

例えば正解が“１２３４”のとき

プレーヤが回答した数字が，

《第13-1表》

プレーヤの回答(入力)	HIT	BLOW
1　2　4　3	2	2
0　1　2　4	1	2
5　6　7　8	0	0
4　3　2　1	0	4
1　2　3　4	4	0

「正解１２３４の場合」

１２４３とすると　この場合HIT＝2，BLOW＝2となり

０１２４とすると　この場合HIT＝1，BLOW＝2となり

５６７８とすると　HIT＝0，BLOW＝0であり

４３２１とすると　HIT＝0，BLOW＝4となります。

つまり正解を当てたときはHIT＝4のときになります。このように数字を入力していってそれまでに入れた数字のときのHIT数とBLOW数を参考にしながら何回で正解を当てられるかを競います **(第13―1表)**。

概略フローチャート

第13―1図に概略フローチャートを示します。

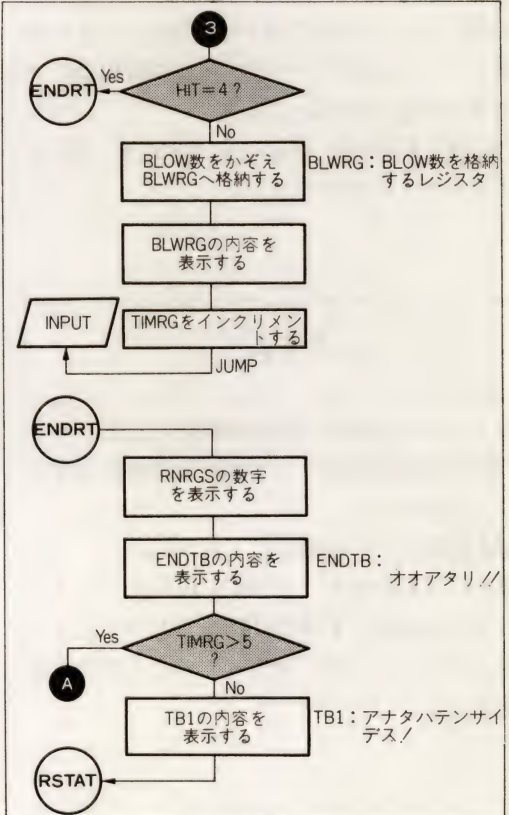
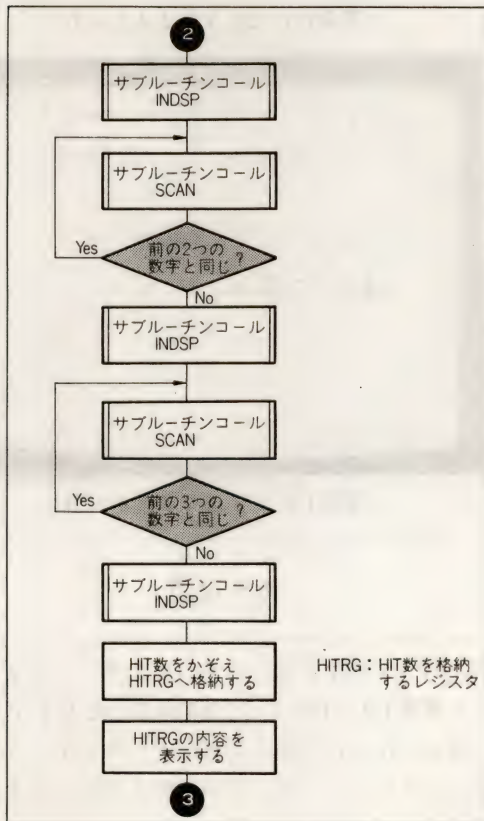
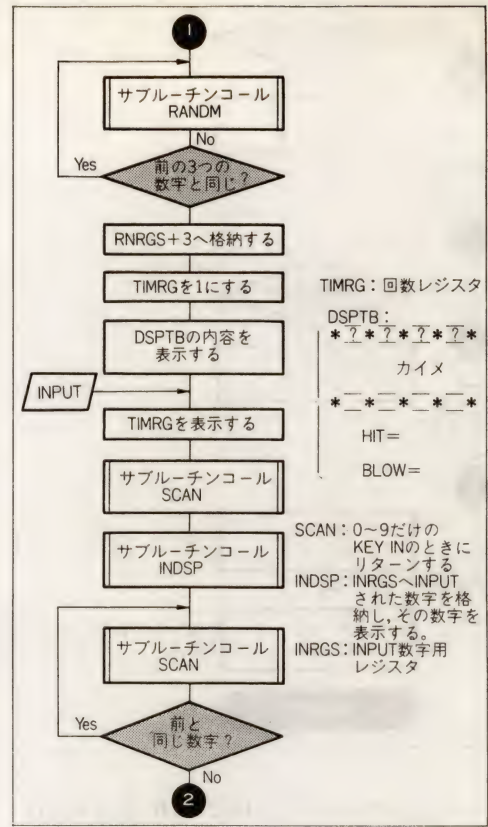
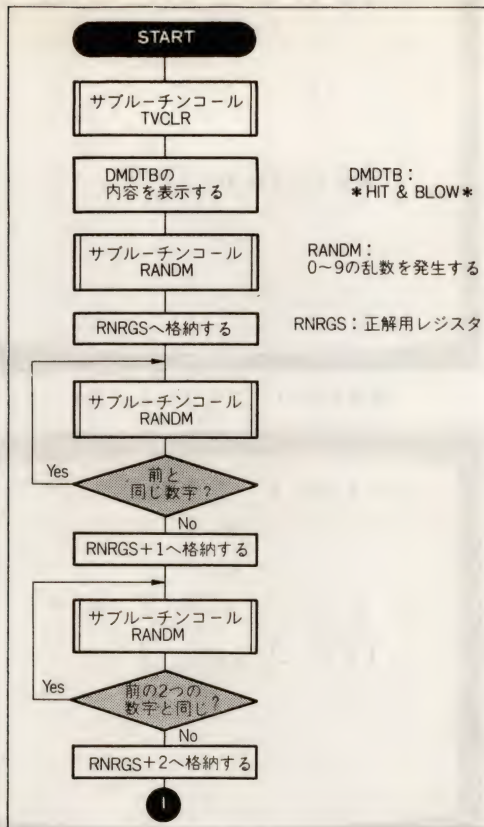
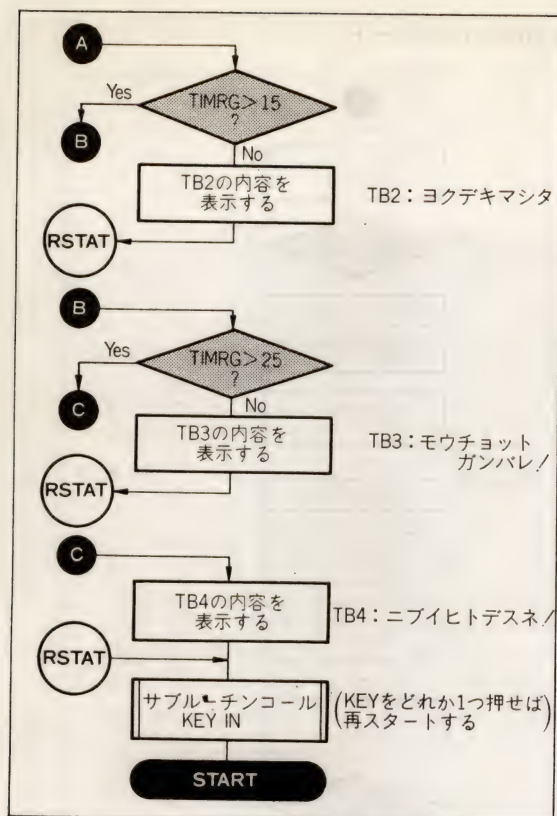
このフローチャートはゲームのプロセスにほぼ合せて書いてありますので，すぐおわかりいただけるものと思います。

サブルーチンプログラム

本プログラムには次のサブルーチンプログラムを使用しておりますので簡単に説明しておきます。

RANDM：０～９までの乱数を発生します。

SCAN　：０～９までのKEYが押されたらリ

《第13—1図》HIT&BLOWフローチャート

ターンします(A～F, RDC, WIC等のキーは受付けない)

INDSP：入力された数をINRGSレジスタに格納し，その数をTVへ出力しますINRGS：入力された数の格納レジスタ。

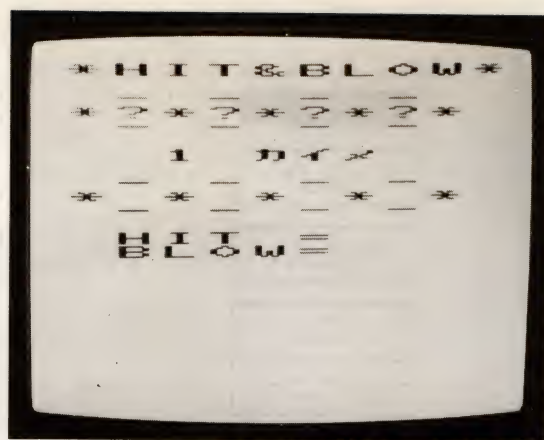
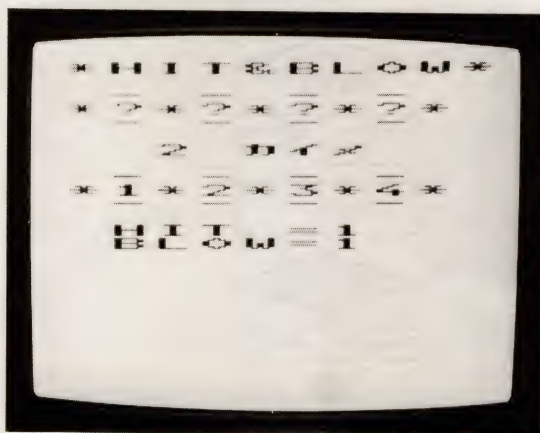
TVCLR, TVDSP, KEYIN, DISPX：EX-80のモニタサブルーチンです。

システム

プログラムの先頭番地を8400HにしてありますのでEX-80の標準構成ではプログラムを収容することはできません。

RAMを1Kバイト増設してください。

具体的にはEX-80ボードのIC9, IC10のエリアにソケットを取り付けTMM314Pをさしてください。

電源，テレビは今まで使用していたもので支障ありません。

《写真13—1》プログラム・スタート

《写真13—2》 1 2 3 4と入力

《写真13—3》 5 6 7 8と入力

あそび方

1．8400番地からプログラムをスタートさせますと**写真13—1**のようなTV画面になります。答えが作られ，かくされたところです。
2．つぎにキー・ボードから答えを入れます。

《写真13—4》ハズレ

《写真13—5》正解

写真13—2は“１２３４”と入れた場合で，HIT＝0，BLOW＝1と表示されています。

3．さて次のキー・インをする前に記憶力に自信のない人はメモをとっておくとよいでしょう。

4．次に“5678”とキー・インしてみました（**写真13—3**）。

5．4桁の数字をつぎつぎと入力します。

6．11回目でHIT＝4，BLOW＝0となり当りました。“ニブイヒトデスネ！：にはがっくりというところです（**写真13—5**）。

さて，ここでHIT & BLOWのMASTER（EX-80）が作り出す数（MIND）は何通りあるか検討してみますと，0～9までの10個の数字のうち同じ数字を使わない4桁の数ですから“X＝10×9×8×7＝5040とおり”となります。このうちの1組を当てるわけですから，盲滅法にやってもまぐれであたることもありますが，定石があります。

第一回目は“０１２３”

二回目は“４５６７”

三回目は“０４８９”

としてHITとBLOWの数をみながら4回目の数をキー・インします。この方法をとりますと7回位で当てることができるでしょう。

《第13—2表》プログラム・リスト

```
                 H I T   &   B L O W
8400                             ORG    8400H
8400    CD6C03      START:       CALL   TVCLR
8403    212386                   LXI    H,DMDTB
8406    060A                     MVI    B,0AH
8408    CD0486                   CALL   DSPLY
840B    21AF86                   LXI    H,RNRGS
840E    CDDD85                   CALL   RANDM
8411    77                       MOV    M,A
8412    CDDD85      LOOP1:       CALL   RANDM
8415    BE                       CMP    M
8416    CA1284                   JZ     LOOP1
8419    23                       INX    H
841A    77                       MOV    M,A
841B    CDDD85      LOOP2:       CALL   RANDM
841E    BE                       CMP    M
841F    CA1B84                   JZ     LOOP2
8422    2B                       DCX    H
8423    BE                       CMP    M
8424    23                       INX    H
8425    CA1B84                   JZ     LOOP2
8428    23                       INX    H
8429    77                       MOV    M,A
842A    CDDD85      LOOP3:       CALL   RANDM
842D    BE                       CMP    M
842E    CA2A84                   JZ     LOOP3
8431    2B                       DCX    H
8432    BE                       CMP    M
8433    23                       INX    H
8434    CA2A84                   JZ     LOOP3
8437    2B                       DCX    H
8438    2B                       DCX    H
8439    BE                       CMP    M
843A    23                       INX    H
843B    23                       INX    H
843C    CA2A84                   JZ     LOOP3
843F    23                       INX    H
8440    77                       MOV    M,A
8441    3E01                     MVI    A,01H
8443    32B786                   STA    TIMRG
8446    213E80                   LXI    H,DISPX
8449    3601                     MVI    M,01H
844B    23                       INX    H
844C    3602                     MVI    M,02H
844E    EB                       XCHG
844F    212D86                   LXI    H,DSPTB
8452    061B                     MVI    B,1BH
8454    CD0486                   CALL   DSPLY
8457    EB                       XCHG
8458    3606                     MVI    M,06H
```

```
845A    2B                 DCX  H
845B    36Ø3               MVI  M,Ø3H
845D    EB                 XCHG

845E    Ø6Ø4               MVI  B,Ø4H
846Ø    CDØ486             CALL DSPLY
8463    EB                 XCHG
8464    36Ø1               MVI  M,Ø1H
8466    23                 INX  H
8467    36Ø8               MVI  M,Ø8H
8469    EB                 XCHG
846A    Ø61B               MVI  B,1BH
846C    CDØ486             CALL DSPLY
846F    EB                 XCHG
847Ø    36ØC               MVI  M,ØCH
8472    2B                 DCX  H
8473    36Ø1               MVI  M,Ø1H
8475    EB                 XCHG
8476    Ø6ØF               MVI  B,ØFH
8478    CDØ486             CALL DSPLY
847B    213E8Ø    INPUT:   LXI  H,DISPX
847E    3AB786             LDA  TIMRG
8481    FEØA               CPI  ØAH
8483    DA9C84             JC   SKIP1
8486    E6FØ               ANI  ØFØH
8488    ØF                 RRC
8489    ØF                 RRC
848A    ØF                 RRC
848B    ØF                 RRC
848C    C63Ø               ADI  3ØH
848E    36Ø1               MVI  M,Ø1H
849Ø    23                 INX  H
8491    36Ø6               MVI  M,Ø6H
8493    CDDEØ2             CALL TVDSP
8496    3AB786             LDA  TIMRG
8499    E6ØF               ANI  ØFH
849B    2B                 DCX  H
849C    36Ø2      SKIP1:   MVI  M,Ø2H
849E    23                 INX  H
849F    36Ø6               MVI  M,Ø6H
84A1    C63Ø               ADI  3ØH
84A3    CDDEØ2             CALL TVDSP

84A6    36Ø9               MVI M,Ø9H
84A8    2B                 DCX  H
84A9    36ØØ               MVI  M,Ø
84AB    11B386             LXI  D,INRGS
84AE    CDØE86             CALL SCAN
84B1    CD1B86             CALL INDSP
84B4    CDØE86    LOOPA:   CALL SCAN
84B7    EB                 XCHG
84B8    BE                 CMP  M
84B9    EB                 XCHG
84BA    CAB484             JZ   LOOPA
84BD    13                 INX  D
84BE    CD1B86             CALL INDSP
84C1    CDØE86    LOOPB:   CALL SCAN
84C4    EB                 XCHG
84C5    BE                 CMP  M
84C6    EB                 XCHG
84C7    CAC184             JZ   LOOPB
84CA    EB                 XCHG
84CB    2B                 DCX  H
84CC    BE                 CMP  M
84CD    23                 INX  H
84CE    EB                 XCHG
84CF    CAC184             JZ   LOOPB
84D2    13                 INX  D
84D3    CD1B86             CALL INDSP
84D6    CDØE86    LOOPC:   CALL SCAN
84D9    EB                 XCHG
84DA    BE                 CMP  M
84DB    EB                 XCHG
84DC    CAD684             JZ   LOOPC
84DF    EB                 XCHG
84EØ    2B                 DCX  H
84E1    BE                 CMP  M
84E2    23                 INX  H
84E3    EB                 XCHG
84E4    CAD684             JZ   LOOPC
84E7    EB                 XCHG
84E8    2B                 DCX  H
84E9    2B                 DCX  H
84EA    BE                 CMP  M
84EB    23                 INX  H
84EC    23                 INX  H
84ED    EB                 XCHG
84EE    CAD684             JZ   LOOPC
84F1    13                 INX  D
84F2    CD1B86             CALL INDSP
84F5    21AF86             LXI  H,RNRGS
84F8    11B386             LXI  D,INRGS
84FB    AF                 XRA  A
84FC    47                 MOV  B,A
84FD    ØEØ4               MVI  C,Ø4H
84FF    1A        LOOP4:   LDAX D
85ØØ    BE                 CMP  M
85Ø1    C2Ø585             JNZ  SKIP2
```

```
85Ø4    Ø4                 INR  B
85Ø5    23        SKIP2:   INX  H
85Ø6    13                 INX  D
85Ø7    ØD                 DCR  C
85Ø8    C2FF84             JNZ  LOOP4
85ØB    78                 MOV  A,B
85ØC    32B886             STA  HITRG
85ØF    213E8Ø             LXI  H,DISPX
8512    36Ø6               MVI  M,Ø6H
8514    23                 INX  H
8515    36ØC               MVI  M,ØCH
8517    C63Ø               ADI  3ØH
8519    CDDEØ2             CALL TVDSP
851C    3AB886             LDA  HITRG
851F    FEØ4               CPI  Ø4H
8521    CA6785             JZ   ENDRT
8524    AF                 XRA  A
8525    32B986             STA  BLWRG
8528    21AF86             LXI  H,RNRGS
852B    Ø6Ø4               MVI  B,Ø4H
852D    11B386    LOOP5:   LXI  D,INRGS
853Ø    ØEØ4               MVI  C,Ø4H
8532    79        LOOP6:   MOV  A,C
8533    B8                 CMP  B
8534    CA4285             JZ   SKIP3
8537    1A                 LDAX D
8538    BE                 CMP  M
8539    C24285             JNZ  SKIP3
853C    E5                 PUSH H
853D    21B986             LXI  H,BLWRG
854Ø    34                 INR  M
8541    E1                 POP  H
8542    13        SKIP3:   INX  D
8543    ØD                 DCR  C
8544    C23285             JNZ  LOOP6
8547    23                 INX  H
8548    Ø5                 DCR  B
8549    C22D85             JNZ  LOOP5
854C    213E8Ø             LXI  H,DISPX
854F    36Ø6               MVI  M,Ø6H
8551    23                 INX  H
8552    36ØD               MVI  M,ØDH
8554    3AB986             LDA  BLWRG
8557    C63Ø               ADI  3ØH
8559    CDDEØ2             CALL TVDSP
855C    21B786             LXI  H,TIMRG
855F    7E                 MOV  A,M
856Ø    C6Ø1               ADI  Ø1H
8562    27                 DAA
8563    77                 MOV  M,A
8564    C37B84             JMP  INPUT
8567    36ØD      ENDRT:   MVI  M,ØDH
8569    2B                 DCX  H
856A    36Ø6               MVI  M,Ø6H
856C    3E3Ø               MVI  A,3ØH
856E    CDDEØ2             CALL TVDSP
8571    36Ø1               MVI  M,Ø1H
8573    23                 INX  H
8574    36Ø3               MVI  M,Ø3H
8576    11AF86             LXI  D,RNRGS
8579    Ø6Ø4               MVI  B,Ø4H
857B    2B                 DCX  H
857C    1A        LOOP7:   LDAX D
857D    C63Ø               ADI  3ØH
857F    CDDEØ2             CALL TVDSP
8582    34-                INR  M
8583    13                 INX  D
8584    Ø5                 DCR  B
8585    C27C85             JNZ  LOOP7
8588    36ØØ               MVI  M,Ø
858A    23                 INX  H
858B    361Ø               MVI  M,1ØH
858D    EB                 XCHG
858E    217686             LXI  H,ENDTB
8591    Ø6Ø8               MVI  B,Ø8H
8593    CDØ486             CALL DSPLY
8596    3AB786             LDA  TIMRG
8599    EB                 XCHG
859A    3613               MVI  M,13H
859C    2B                 DCX  H
859D    36ØØ               MVI  M,Ø
859F    FEØ5               CPI  Ø5H
85A1    D2AF85             JNC  SKIP4
85A4    217E86             LXI  H,TB1
85A7    Ø6ØE               MVI  B,ØEH
85A9    CDØ486             CALL DSPLY
85AC    C3D785             JMP  RSTAT
85AF    FE15      SKIP4:   CPI  15H
85B1    D2BF85             JNC  SKIP5
85B4    218C86             LXI  H,TB2

85B7    Ø6Ø9               MVI  B,Ø9H
85B9    CDØ486             CALL DSPLY
85BC    C3D785             JMP  RSTAT
85BF    FE25      SKIP5:   CPI  25H
85C1    D2CF85             JNC  SKIP6
85C4    219586             LXI  H,TB3
85C7    Ø6ØE               MVI  B,ØEH
85C9    CDØ486             CALL DSPLY
85CC    C3D785             JMP  RSTAT
```

```
85CF  21A386   SKIP6:  LXI  H,TB4
85D2  060C             MVI  B,0CH
85D4  CD0486           CALL DSPLY
85D7  CD4402   RSTAT:  CALL KEYIN
85DA  C30084           JMP  START
 S U B R O U T I N E   R A N D O M
               ;
85DD  E5       RANDM:  PUSH H
85DE  2ABA86           LHLD RNDPT
85E1  3ABC86           LDA  RNDRG
85E4  AE               XRA  M
85E5  E607             ANI  07H
85E7  67               MOV  H,A
85E8  AE               XRA  M
85E9  85               ADD  L
85EA  6F               MOV  L,A
85EB  22BA86           SHLD RNDPT
85EE  3ABC86           LDA  RNDRG
85F1  AE       RND1:   XRA  M
85F2  32BC86           STA  RNDRG
85F5  E60F             ANI  0FH
85F7  FE0A             CPI  0AH
85F9  DA0286           JC   RND2
85FC  0F               RRC
85FD  84               ADD  H
85FE  6F               MOV  L,A
85FF  C3F185           JMP  RND1
8602  E1       RND2:   POP  H
8603  C9               RET
               ;
S U B R O U T I N E   D I S P L A Y
               ;
8604  7E       DSPLY:  MOV  A,M
8605  CDDE02           CALL TVDSP
8608  23               INX  H
8609  05               DCR  B
860A  C20486           JNZ  DSPLY
860D  C9               RET
S U B R O U T I N E   K E Y   S C A N
               ;
860E  E5       SCAN:   PUSH H
860F  D5               PUSH D
8610  CD4402           CALL KEYIN
8613  FE0A             CPI  0AH
8615  D21086           JNC  $-5
8618  D1               POP  D
8619  E1               POP  H
861A  C9               RET
  D A T A   D I S P L A Y
861B  12       INDSP:  STAX D
861C  C630             ADI  30H
861E  34               INR  M
861F  CDDE02           CALL TVDSP
8622  C9               RET
```

```
                        ;       T A B L E
                        ;
8623  2A080914 DMDTB:   DB    2AH,08H,09H,14H
8627  26020C0F          DB    26H,02H,0CH,0FH
862B  172A              DB    17H,2AH
862D  2D202D20 DSPTB:   DB    2DH,20H,2DH,20H
8631  2D202D20          DB    2DH,20H,2DH,20H
8635  202A3F2A          DB    20H,2AH,3FH,2AH
8639  3F2A3F2A          DB    3FH,2AH,3FH,2AH
863D  3F2A2020          DB    3FH,2AH,20H,20H
8641  2D202D20          DB    2DH,20H,2DH,20H
8645  2D202D            DB    2DH,20H,2DH
8648  20767252          DB    20H,76H,72H,52H
864C  2D202D20          DB    2DH,20H,2DH,20H
8650  2D202D20          DB    2DH,20H,2DH,20H
8654  202A202A          DB    20H,2AH,20H,2AH
8658  202A202A          DB    20H,2AH,20H,2AH
865C  202A2020          DB    20H,2AH,20H,20H
8660  2D202D20          DB    2DH,20H,2DH,20H
8664  2D202D            DB    2DH,20H,2DH
8667  08091420          DB    08H,09H,14H,20H
866B  3D202020          DB    3DH,20H,20H,20H
866F  2020020C          DB    20H,20H,02H,0CH
8673  0F173D            DB    0FH,17H,3DH
8676  75757140 ENDTB:   DB    75H,75H,71H,40H
867A  58202121          DB    58H,20H,21H,21H
867E  71454020 TB1:     DB    71H,45H,40H,20H
8682  4A20435D          DB    4AH,20H,43H,5DH
8686  7B72435E          DB    7BH,72H,43H,5EH
868A  7D21              DB    7DH,21H
868C  5678435E TB2:     DB    56H,78H,43H,5EH
8690  774F7C40          DB    77H,4FH,7CH,40H
8694  61                DB    61H
8695  5373416E TB3:     DB    53H,73H,41H,6EH
8699  6F442076          DB    6FH,44H,20H,76H
869D  5E5D4A5E          DB    5EH,5DH,4AH,5EH
86A1  5A21              DB    5AH,21H
86A3  464C5E72 TB4:     DB    46H,4CH,5EH,72H
86A7  4B442043          DB    4BH,44H,20H,43H
86AB  5E7D4821          DB    5EH,7DH,48H,21H
                        ;
86AF                            ORG   86AFH
                        ;
                        ;       W O R K   A R E A
0004                    RNRGS:  DS    4
0004                    INRGS:  DS    4
0001                    TIMRG:  DS    1
0001                    HITRG:  DS    1
0001                    BLWRG:  DS    1
0002                    RNDPT:  DS    2
0001                    RNDRG:  DS    1
                        ;
036C                    TVCLR   EQU   036CH
02DE                    TVDSP   EQU   02DEH
0244                    KEYIN   EQU   0244H
803E                    DISPX   EQU   803EH
                        ;
                                END
```

SYMBOL TABLE

```
* 01

A       0007     B       0000     BLWRG   86B9     C       0001
D       0002     DISPX   803E     DMDTB   8623     DSPLY   8604
DSPTB   862D     E       0003     ENDRT   8567     ENDTB   8676
H       0004     HITRG   86B8     INDSP   861B     INPUT   847B
INRGS   86B3     KEYIN   0244     L       0005     LOOP1   8412
LOOP2   841B     LOOP3   842A     LOOP4   84FF     LOOP5   852D
LOOP6   8532     LOOP7   857C     LOOPA   84B4     LOOPB   84C1
LOOPC   84D6     M       0006     PSW     0006     RANDM   85DD
RND1    85F1     RND2    8602     RNDPT   86BA     RNDRG   86BC
RNRGS   86AF     RSTAT   85D7     SCAN    860E     SKIP1   849C
SKIP2   8505     SKIP3   8542     SKIP4   85AF     SKIP5   85BF
SKIP6   85CF     SP      0006     START   8400     TB1     867E
TB2     868C     TB3     8695     TB4     86A3     TIMRG   86B7
TVCLR   036C     TVDSP   02DE
```

14

カナ文字の利用法

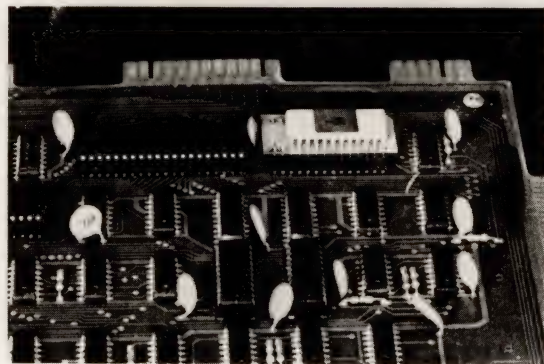
百人一首プログラム

PET2001

劔持　甫

コモドール社では，国内のPETのユーザー向けに，先ごろカナROMを発売しましたので，さっそく私も入手使用してみました。

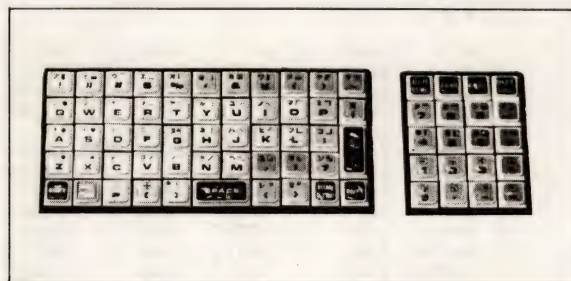
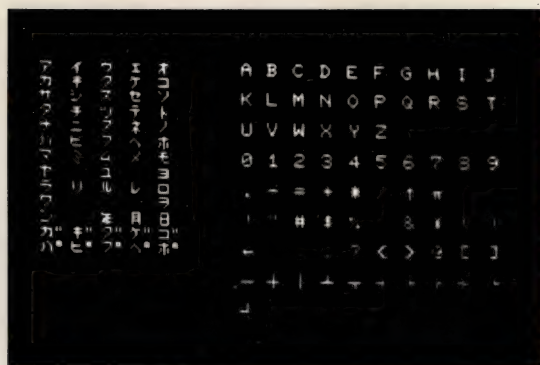
プログラムは，カナ文字でなければならないものと考えて，優雅に百人一首を読んでみることにしました。

カナROM

PETは，いままで英大文字，グラフィック文字，数字およびPOKE文で英小文字が使えました。

発売されたカナROMは，**写真14—1**のように従来のキャラクターゼネレーター（MPS6540）を取りはずして差替えればよいようになっており，使用するときはPOKE59468,14で英小文字の代りに使うことができます。（これによって，英小文字および一部のグラフィック文字が使えなくなります。）また，キーボードは，キートップにカナ文字が入ったものが新しくカナROMに附属されていますので，これに取り替える必要があります。

《写真14—1》キャラクタ・ジェネレータを取替え

新しいキー・ボードは，**写真14—2**のようになります。

また，ディスプレイしたPETのカナ文字およびカナ文字使用時に使用できる文字は**写真14—3**のとおりです。

《写真14—2》キー・トップを取替え

《写真14―3》新しく画面にでる文字

プログラムの概要

百人一首は，ご存知のとおり百首の上の句と下の句でできており，両者を組合せるものです。

ここでは，簡単に上の句と下の句を並べてディスプレイするようにしました。次のステップとして，「カルタ取り」の形式にできるようにしてあります。

百首の句を，1本のプログラムにすると8Kのメモリーを超えてしまいますので，ここでは，50首ずつに分けてプログラムし，プログラムの中でそれぞれのプログラムを接続しています。

このプログラムは，前の50首と後の50首に分けられそれぞれは次の五つのブロックで構成されています。

上の句の選定

このブロックは，ラインナンバー300～490で構成されています。

上の句の選定は，ラインナンバー320によって，乱数を発生させその中の1～50の数を利用しています。

ここで，選ばれた1～50の乱数によってラインナンバー430～439からON…GOTO文により上の句のブロックへジャンプします。また，ラインナンバー490は，前の50首が終了したときに後の50首に移るために，次のテープをロードするためのものです。

上の句

ラインナンバー1500～1745には，上の句（5・7・5）がA$に代入されています。

上の句のプリント

このブロックは，ラインナンバー500～995で構成されています。

ラインナンバー1500以降でA$に代入された上の句は，1句ごとに1ラインに書かれていますが，文学的表現の日本語（特にカナ文字）は横に書くと読みにくい欠点があります。

そこで，ここでは，A$に代入された横書きの文字を縦書きになおす「クロス交換」を行なっています。

特に，濁点がある時には，変換時に注意を必要とします。

下の句の選定およびプリント

このブロックは，ラインナンバー1000～1440で構成されています。

ここでは，AD$に代入された下の句を，上の句の選定に利用した結果の乱数によりON…GOSUB文によって下の句のブロックにジャンプさせています。

プリントは，上の句のプリントと同時にAD$の横書き文を縦書き文に「クロス交換」しています。

なお，単に上の句と下の句をプリントするだけならば，A$とAD$に分ける必要はありませんが，将来，上の句をディスプレイしてそれに対する下の句を「取る」ようにグレードアップするために分けてあります。

下の句

ラインナンバー2000～2245には，下の句（7・7）がAD$に代入されています。

では，次にラインナンバーを追って，各ブロックごとに説明しましょう。

前の50首のプログラム

上の句の選定

プログラムは，**写真14―4**のとおりです。

上の句の選定は，前に説明したように乱数を利用しています。

```
10 REM "** ヒヤクニン イッシュ **"
20 POKE 59468,14
30 PRINT "■"
40 PRINT"■■■■■■■■■■■■■■■■■■■■■■ヒヤクニン イッシュ"
50 PRINT"■■■■■■■■■■■■■■■■■■■マイ 50 シュ"
60 TM=TI
70 IF (TI-TM)<500 THEN 70
80 PRINT"■"
300 DIM H(50):S=1
310 P=P+1:IF P=51 THEN 460
315 AU=1
320 I=INT(RND(1)*100 )
330 IF I=0 THEN 320
340 IF I>51 THEN 320
350 IF P=1 THEN H(1)=I
360 IF P=1 THEN 429
370 FOR W=1 TO S:IFH(W)=ITHENAU=0:NEXTW
380 IF AU=0 THEN 315
390 S=S+1
400 H(S)=I
410 H0=0:H1=0:H2=0:H3=0:H4=0
420 IF H(S)>40 THEN 438
421 IF H(S)>30 THEN 436
422 IF H(S)>20 THEN 434
423 IF H(S)>10 THEN 432
429 H0=H(S)
430 ON H0 GOTO 1500,1505,1510,1515,1520,1525,1530,1535,1540,1545
432 H1=H(S)-10
433 ON H1 GOTO 1550,1555,1560,1565,1570,1575,1580,1585,1590,1595
434 H2=H(S)-20
435 ON H2 GOTO 1600,1605,1610,1615,1620,1625,1630,1635,1640,1645
436 H3=H(S)-30
437 ON H3 GOTO 1650,1655,1660,1665,1670,1675,1680,1685,1690,1695
438 H4=H(S)-40
439 ON H4 GOTO 1700,1705,1710,1715,1720,1725,1730,1735,1740,1745
450 GOTO 310
460 PRINT"■"
470 PRINT"■■■■■■■■■■■■■■■■■■■■マイ 50 シュ ノ オワリ"
480 PRINT"■■■■■■■■■■アト 50 シュ ヲ TAPE カラ LOAD サレマス■■■■■■■■■■"
490 CLR:LOAD:END
```

《写真14—4》上の句の選定

ラインナンバー300は, 50首に相当する乱数を入れるための配列H (50) を宣言しています。

ラインナンバー320は, 使用のつどに新しい乱数を発生させ, その数を100倍して0～99の乱数を作り, 変数 I に代入しています。

ラインナンバー330, 340は, 0～99の乱数の中から1～50の乱数を選び出しています。

ラインナンバー350は, 配列の1番目に最初に発生した必要な乱数を入れています。

ラインナンバー370は, 配列の中に同じ数が入らないようにするための比較を行なっています。

ラインナンバー400は, ラインナンバー370によって選別された同じ数がない1～50の数が入ります。そのカウンターは, ラインナンバー390です。

ラインナンバー420～423は, 上の句を取り出しにいくON…GOTO文のジャンプ指定が100個できればいらないのですが, 指定できるジャンプ先が10個の為11～50の数を10単位でそれぞれ1～10に変換するためのものです。例えば, 乱数が34の場合には, ラインナンバー421によってラインナンバー436にジャンプし, ラインナンバー437のH3が4となりラインナンバー1665へジャンプします。

ラインナンバー429～439は, 上の句へジャンプさせるON…GOTO文です。

ラインナンバー460～480は, 前の50首の終わりを示すプリント文です。

ラインナンバー490は, 次のテープをロードしランさせるもので, このラインの実行によって次のプログラムが自動的にロードされて実行を開始するので, 実行的には1本のプログラムのようにみることができます。

上の句

プログラムは, **写真14—5**のとおりです。

上の句は, ラインナンバー1500～1745のA$に代入されています。

上の句のプリント

プログラムは, **写真14—6**のとおりです。

ここでA$に書かれた横文字を縦にプリントするためのクロス変換を行なっています。

A$に代入された上の句は, ラインナンバー500にジャンプされます。

ラインナンバー530は, 上の句を縦に23文字プリントします。５・７・５で17文字なのになぜ23文字必要かは, 濁点があるためです。

ラインナンバー508と995は, 札の上および下の横線をプリントします。

ラインナンバー540～780は, クロス変換を行なっています。

第14—1図の例を掲げて説明します。

ラインナンバー1740のA$には, キミガタメ……が代入されています。

ラインナンバー1740からラインナンバー500にジャンプすると, ラインナンバー540のB$には**キミ**という2文字が代入され, D$には**ミ**が, I$には**キ**が代入されますので, ラインナンバー620および640によって画面の上から**第14—1図**のようにプリントされます。

次にB$には**カ〟**が代入され, D$には〟が代入されますからラインナンバー670のG$に代入された**カ**をラインナンバー680でプリントし, それに続けて〟をプリントします。

このように, 濁点も1文字として数えて2文字ずつ処理していくと, 7回目の処理ではB$に**〟リ**が代

```
1500 A$= "ﾜﾀﾉﾊﾗ ﾔｿｼﾏｶｹﾃ ｺｷﾞｲﾃﾞﾇ    ":GOTO 500
1505 A$= "ﾊﾙｽｷﾞﾃ ﾅﾂｷﾆｹﾗｼ ｼﾛﾀｴﾉ    ":GOTO 500
1510 A$="ｱﾏﾂｶｾﾞ ｸﾓﾉｶﾖｲｼﾞ ﾌｷﾄｼﾞﾖ   ":GOTO 500
1515 A$="ﾂｸﾊﾞﾈﾉ ﾐﾈﾖﾘｵﾂﾙ ﾐﾅﾉｶﾜ    ":GOTO 500
1520 A$="ﾐﾁﾉｸﾉ ｼﾉﾌﾞﾓﾁﾞｽﾞﾘﾀﾞﾚﾕｴﾆ  ":GOTO 500
1525 A$="ｷﾐｶﾞﾀﾒ ﾊﾙﾉﾉﾆｲﾃﾞﾃ ﾜｶﾅﾂﾑ  ":GOTO 500
1530 A$="ﾀﾁﾜｶﾚ ｲﾅﾊﾞﾉﾔﾏﾉ ﾐﾈﾆｳﾏﾌﾙ  ":GOTO 500
1535 A$="ﾁﾊﾔﾌﾞﾙ ｶﾐﾖﾓｷｶｽﾞ ﾀﾂﾀｶﾞﾜ  ":GOTO 500
1540 A$="ｽﾐﾉｴﾉ ｷｼﾆﾖﾙﾅﾐ ﾖﾙｻｴﾔ     ":GOTO 500
1545 A$="ﾅﾆﾜｶﾞﾀ ﾐｼﾞｶｷﾛﾉ ﾌｼﾉﾏﾓ   ":GOTO 500
1550 A$="ｱｷﾉﾀﾉ ｶﾘﾎﾉｲﾎﾉ ﾄﾏｦｱﾗﾐ    ":GOTO 500
1555 A$="ｱｼﾋｷﾉ ﾔﾏﾄﾞﾘﾉｵﾉ ｼﾀﾞﾘｵﾉ  ":GOTO 500
1560 A$="ﾀｺﾞﾉｳﾗﾆｳﾁｲﾃﾞﾃﾐﾚﾊﾞｼﾛﾀｴﾉ  ":GOTO 500
1565 A$="ｵｸﾔﾏﾆ ﾓﾐｼﾞﾌﾐﾜｹ ﾅｸｼｶﾉ   ":GOTO 500
1570 A$="ｶｻｻｷﾞﾉ ﾜﾀｾﾙﾊｼﾆ ｵｸｼﾓﾉ   ":GOTO 500
1575 A$="ﾜｶﾞｲﾎﾊ ﾐﾔｺﾉﾀﾂﾐ ｼｶｿﾞｽﾑ  ":GOTO 500
1580 A$="ﾊﾅﾉｲﾛﾊ ｳﾂﾘﾆｹﾘﾅ ｲﾀﾂﾞﾗﾆ  ":GOTO 500
1585 A$="ｺﾚﾔｺﾉ ﾕｸﾓｶｴﾙﾓ ﾜｶﾚﾃﾊ    ":GOTO 500
1590 A$="ﾜﾋﾞﾇﾚﾊﾞ ｲﾏﾊﾀｵﾅｼﾞ ﾅﾆﾜﾅﾙ ":GOTO 500
1595 A$="ｲﾏｺﾑﾄ ｲﾋｼﾊﾞｶﾘﾆ ﾅｶﾞﾂｷﾉ  ":GOTO 500
1600 A$="ﾌｸｶﾗﾆ ｱｷﾉｸｻｷﾉ ｼｦﾙﾚﾊﾞ   ":GOTO 500
1605 A$="■ﾐﾚﾊﾞ ﾁﾁﾞﾆﾓﾉｺｿ ｶﾅｼｹﾚ  ":GOTO 500
1610 A$="ｺﾉﾀﾋﾞﾊ ﾇｻﾓﾄﾘｱｴｽﾞ ﾀﾑｹﾔﾏ ":GOTO 500
1615 A$="ﾅﾆｵﾊﾊﾞ ｱｲｻｶﾔﾏﾉ ｻﾈｶﾂﾞﾗ  ":GOTO 500
1620 A$="ｦｸﾞﾗﾔﾏ ﾐﾈﾉﾓﾐｼﾞﾊﾞｺｺﾛｱﾗﾊﾞ":GOTO 500
1625 A$="ﾐｶﾉﾊﾗ ﾜｷﾃﾅｶﾞﾙﾙ ｲｽﾞﾐｶﾞﾜ ":GOTO 500
1630 A$="ﾔﾏｻﾄﾊ ﾌﾕｿﾞｻﾋﾞｼｻ ﾏｻﾘｹﾙ  ":GOTO 500
1635 A$="ｺｺﾛｱﾃﾆ ｵﾗﾊﾞﾔｵﾗﾑ ﾊﾂｼﾓﾉ  ":GOTO 500
1640 A$="ｱﾘｱｹﾉ ﾂﾚﾅｸﾐｴｼ ﾜｶﾚﾖﾘ    ":GOTO 500
1645 A$="ｱｻﾎﾞﾗｹ ｱﾘｱｹﾉﾂｷﾄ ﾐﾙﾏﾃﾞﾆ ":GOTO 500
1650 A$="ﾔﾏｶﾞﾊﾆ ｶｾﾞﾉｶｹﾀﾙ ｼｶﾞﾗﾐﾊ ":GOTO 500
1655 A$="ﾋｻｶﾀﾉ ﾋｶﾘﾉﾄﾞｹｷ ﾊﾙﾉﾋﾆ   ":GOTO 500
1660 A$="ﾀﾞﾚｦｶﾓ ｼﾙﾋﾄﾆｾﾑ ﾀｶｻｺﾞﾉ  ":GOTO 500
1665 A$="ﾋﾄﾊｲｻ ｺｺﾛﾓｼﾗｽﾞ ﾌﾙｻﾄﾊ   ":GOTO 500
1670 A$="ﾅﾂﾉﾖﾊ ﾏﾀﾞﾖｲﾅｶﾞﾗ ｱｹﾇﾙｦ  ":GOTO 500
1675 A$="ｼﾗﾂﾕﾆ ｶｾﾞﾉﾌｷｼｸ ｱｷﾉﾉﾊ   ":GOTO 500
1680 A$="ﾜｽﾗﾙﾙ ﾐｦﾊﾞｵﾓﾊｽﾞ ﾁｶﾋﾃｼ  ":GOTO 500
1685 A$="ｱｻｶﾞﾔﾉ ｵﾉﾉｼﾉﾊﾗ ｼﾉﾌﾞﾚﾄﾞ ":GOTO 500
1690 A$="ｼﾉﾌﾞﾚﾄﾞｲﾛﾆｲﾃﾞﾆｹﾘﾜｶﾞｺｲﾊ ":GOTO 500
1695 A$="ｺｲｽﾃﾌ ﾜｶﾞﾅﾊﾏﾀﾞｷ ﾀﾁﾆｹﾘ  ":GOTO 500
1700 A$="ﾁｷﾞﾘｷﾅ ｶﾀﾐﾆｿﾃﾞｦ ｼﾎﾞﾘﾂﾂ ":GOTO 500
1705 A$="ｱｲﾐﾃﾉ ｱﾄﾉｺｺﾛﾆ ｸﾗﾌﾞﾚﾊﾞ  ":GOTO 500
1710 A$="ｱﾌｺﾄﾉ ﾀｴﾃｼﾅｸﾊ ﾅｶﾅｶﾆ    ":GOTO 500
1715 A$="ｱﾊﾚﾄﾓ ｲﾌﾍﾞｷﾋﾄﾊ ｵﾓﾎｴﾃﾞ  ":GOTO 500
1720 A$="ﾕﾗﾉﾄｦ ﾜﾀﾙﾌﾅﾋﾞﾄ ｵｼﾞｦﾀｴ  ":GOTO 500
1725 A$="ﾔｴﾑｸﾞﾗ ｼｹﾞﾚﾙﾔﾄﾞﾉｻﾋﾞｼｷﾆ ":GOTO 500
1730 A$="ｶｾﾞｦｲﾀﾐ ｲﾜｳﾂﾅﾐﾉ ｵﾉﾚﾉﾐ  ":GOTO 500
1735 A$="ﾐｶｷﾓﾘ ｴｼﾉﾀｸﾋﾉ ﾖﾙﾊﾓｴ    ":GOTO 500
1740 A$="ｷﾐｶﾞﾀﾒ ｵｼｶﾗｻﾞﾘｼ ｲﾉﾁｻｴ  ":GOTO 500
1745 A$="ｶｸﾄﾀﾞﾆ ｴﾔﾊｲﾌｷﾉ ｻｼﾓｸﾞｻ  ":GOTO 500

READY.
```

《写真14―5》上の句のプログラム

```
LIST500-650
 500 PRINT"■"
 505 C=1
 508 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■──────■"
 510 IF C=1 THEN 540
 520 C=C+2
 530 IF C=23 THEN 995
 540 B$=MID$(A$,C,2):D$=RIGHT$(B$,1)
 560 IF D$="ﾞ" THEN 670
 570 IF D$="ﾟ" THEN 700
 580 I$=LEFT$(B$,1)
 590 IF I$="ﾞ" THEN 740
 600 IF I$="ﾟ" THEN 770
 610 E$=LEFT$(B$,1)
 620 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"E$""■|"
 630 F$=RIGHT$(B$,1)
 640 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"F$""■|"
 650 GOTO 520
 670 G$=LEFT$(B$,1)
 680 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"G$"ﾞ"■|":GOTO650
 700 H$=LEFT$(B$,1)
 710 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"H$"ﾟ"■|":C=C+1:GOTO650
 740 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■■"I$"|"
 750 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"D$"■|":GOTO650
 760 GOTO 650
 770 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■■"I$"■|"
 780 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■|■"D$"|"
 995 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■──────■"

READY.
```

《写真14―6》上の句のプリント

《第14―1図》縦表示のアルゴリズム

入されます。すると，D$には**リ**が，I$には゛が代入されるためにラインナンバー740でカーソルを1ライン上に移動し，さらに6回目にプリントした**サ**の文字の右側に移動してI$つまり゛をプリントします。そして，次の行にラインナンバー750でD$つまり**リ**をプリントします。

このようにして，A$に横に書かれた文字を縦に変換してプリントすると，次は下の句の選定になります。

下の句の選定およびプリント

プログラムは，**写真14―7**のとおりです。

下の句の選定は，上の句の選定のとき使用したH0～H4の値を利用して，ラインナンバー1015

```
1000 REM "**ｼﾓﾉｸ ﾉ ｾﾝﾃｲ**"
1005 PRINT""
1015 ON H0 GOSUB 2000,2005,2010,2015,2020,2025,2030,2035,2040,2045
1018 ON H1 GOSUB 2050,2055,2060,2065,2070,2075,2080,2085,2090,2095
1020 ON H2 GOSUB 2100,2105,2110,2115,2120,2125,2130,2135,2140,2145
1022 ON H3 GOSUB 2150,2155,2160,2165,2170,2175,2180,2185,2190,2195
1024 ON H4 GOSUB 2200,2205,2210,2215,2220,2225,2230,2235,2240,2245
1100 PRINT"":CD=1
1105 PRINT"────"
1110 IF CD=1 THEN 1140
1120 CD=CD+2
1130 IF CD=19 THEN 1400
1140 BD$=MID$(AD$,CD,2)
1150 DD$=RIGHT$(BD$,1)
1160 IF DD$="ﾞ" THEN 1270
1170 IF DD$="ﾟ" THEN 1300
1180 ID$=LEFT$(BD$,1)
1190 IF ID$="ﾞ" THEN 1340
1200 IF ID$="ﾟ" THEN 1370
1210 ED$=LEFT$(BD$,1)
1220 PRINT"|"ED$" |"
1230 FD$=RIGHT$(BD$,1)
1240 PRINT"|"FD$" |"
1250 GOTO 1120
1270 GD$=LEFT$(BD$,1)
1280 PRINT "|"GD$"" |"
1290 GOTO 1250
1300 HD$=LEFT$(BD$,1)
1310 PRINT"|"HD$"" |"
1320 CD=CD+1
1340 PRINT""ID$" |"
1350 PRINT"|"DD$" |"
1360 GOTO 1250
1370 PRINT"|"ID$" |"
1380 PRINT"|"DD$" |"
1390 GOTO 1250
1400 PRINT"────"
1410 TM=TI
1420 IF (TI-TM)<500 THEN 1420
1430 PRINT ""
1440 GOTO 310

READY.
```

《写真14―7》下の句の選定及びプリント

～1024のON…GOSUB文によって行なっています。

また，ラインナンバー1140～1390では，下の句のクロス変換を行なっています。

ラインナンバー1410，1420は，カウンターで上の句と下の句を画面上にディスプレイする時間を決めています。

ラインナンバー1440は，次の句の選定に行くGOTO文です。

下の句

プログラムは，**写真14―8**のとおりです。

下の句は，ラインナンバー2000～2245で，AD$に代入されています。

後50首のプログラム

以上で，前50首のプログラムについての説明を終わります。後50首については，前50首とほとんど同様ですから変更した部分だけを**写真14―9，10**に示します。

```
2000 AD$="ﾅﾎｳﾗﾒｼｷ ｱｻﾎﾞﾗｹｶﾅ  ":RETURN
2005 AD$="ｲｶﾆﾋｻｼｷ ﾓﾉﾄｶﾊｼﾙ  ":RETURN
2010 AD$="ｹﾌﾗｶｷﾞﾘﾉ ｲﾉﾁﾄﾓｶﾞﾅ  ":RETURN
2015 AD$="ﾅｺｿﾅｶﾞﾚﾃ ﾅｦｷｺｴｹﾚ  ":RETURN
2020 AD$="ｲﾏﾋﾄﾀﾋﾞﾉ ｱﾌｺﾄﾓｶﾞﾅ  ":RETURN
2025 AD$="ｸﾓｶﾞｸﾚﾆｼ ﾔﾊﾝﾉﾂｷｶｹﾞ  ":RETURN
2030 AD$="ｲﾃﾞｿﾖﾋﾄｦ ﾜｽﾚﾔﾊｽﾙ  ":RETURN
2035 AD$="ｶﾀﾌﾞｸﾏﾃﾞﾉ 月ｦﾐｼｶﾅ  ":RETURN
2040 AD$="ﾏﾀﾞﾌﾐﾓﾐｽﾞ ｱﾏﾉﾊｼﾀﾞﾃ  ":RETURN
2045 AD$="ｹﾌ ｺｺﾉｴﾆ ﾆｵﾋﾇﾙｶﾅ  ":RETURN
2050 AD$="ﾖﾆｱｲｻﾞｶﾉ ｾｷﾊﾕﾙｻｼﾞ  ":RETURN
2055 AD$="ﾋﾄﾂﾞﾃﾅﾗﾃﾞ ｲﾌﾖｼﾓｶﾞﾅ  ":RETURN
2060 AD$="ｱﾗﾜﾗﾜﾀﾙ ｾｾﾞﾉｱｼﾞﾛｷﾞ  ":RETURN
2065 AD$="ｺｲﾆｸﾁﾅﾑ ﾅｺｿｦｼｹﾚ  ":RETURN
2070 AD$="ﾊﾅﾖﾘﾎｶﾆ ｼﾙﾋﾄﾓﾅｼ  ":RETURN
2075 AD$="ｶﾋﾅｸﾀﾀﾑ ﾅｺｿｵｼｹﾚ  ":RETURN
2080 AD$="ｺｲｼｶﾙﾍﾞｷ ﾔﾊﾝﾉ月ｶﾅ  ":RETURN
2085 AD$="ﾀﾂﾀﾉｶﾜﾉ ﾆｼｷﾅﾘｹﾘ  ":RETURN
2090 AD$="ﾐｦﾂｸｼﾃﾓ ｱﾊﾑﾄｿﾞｵﾓﾌ  ":RETURN
2095 AD$="ｱｼﾉﾏﾛﾔﾆ ｱｷｶｾﾞｿﾞﾌｸ  ":RETURN
2100 AD$="ｶｹｼﾞﾔｿﾃﾞﾉ ﾇﾚﾓｺｿｽﾚ  ":RETURN
2105 AD$="ﾄﾔﾏﾉｶｽﾐ ﾀﾀｽﾞﾓｱﾗﾅﾑ  ":RETURN
2110 AD$="ﾊｹﾞｼｶﾚﾄﾊ ｲﾉﾗﾇﾓﾉｦ  ":RETURN
2115 AD$="ｱﾜﾚｺﾄｼﾉ ｱｷﾓｲﾇﾒﾘ  ":RETURN
2120 AD$="ｸﾓﾏﾆﾏｶﾞﾌ ｵｷﾂｼﾗﾅﾐ  ":RETURN
2125 AD$="ﾜﾚﾃﾓｽｴﾆ ｱﾊﾑﾄｿﾞｵﾓﾌ  ":RETURN
2130 AD$="ｲｸﾖﾈｻﾞﾒﾇ ｽﾏﾉｾｷﾓﾘ  ":RETURN
2135 AD$="ﾓﾚｲｽﾞﾙ月ﾉ ｶｹﾞﾉｻﾔｹｷ  ":RETURN
2140 AD$="ﾐﾀﾞﾚﾃｹｻﾊ ﾓﾉｦｺｿｵﾓﾌ  ":RETURN
2145 AD$="ﾀﾞﾀﾞｱﾘｱｹﾉ 月ｿﾞﾉｺﾚﾙ  ":RETURN
2150 AD$="ｳｷﾆﾀｴﾇﾊ ﾅﾐﾀﾞﾅﾘｹﾘ  ":RETURN
2155 AD$="ﾔﾏﾉｵｸﾆﾓ ｼｶｿﾞﾅｸﾅﾙ  ":RETURN
2160 AD$="ｳｼﾄﾐｼﾖｿﾞ ｲﾏﾊｺｲｼｷ  ":RETURN
2165 AD$="ﾊﾅｿﾞﾑｶｼﾉ ｶﾆﾆﾎｲｹﾙ  ":RETURN
2170 AD$="ｶｺﾁｶｵﾅﾙ ﾜｶﾞﾅﾐﾀﾞｶﾅ  ":RETURN
2175 AD$="ｷﾘﾀﾁﾉﾎﾞﾙ ｱｷﾉﾕﾌｸﾞﾚ  ":RETURN
2180 AD$="ﾐｦﾂｸｼﾃﾔ ｺﾋﾜﾀﾙﾍﾞｷ  ":RETURN
2185 AD$="ｼﾉﾌﾞﾙｺﾄﾉ ﾖﾜﾘﾓｿﾞｽﾙ  ":RETURN
2190 AD$="ﾇﾚﾆｿﾞﾇﾚｼ ｲﾛﾊｶﾊﾗｽﾞ  ":RETURN
2195 AD$="ｺﾛﾓｶﾀｼｷ ﾋﾄﾘｶﾓﾈﾑ  ":RETURN
2200 AD$="ｽｴﾉﾏﾂﾔﾏ ﾅﾐｺｻｼﾞﾄﾊ  ":RETURN
2205 AD$="ｱﾏﾉｦﾌﾞﾈﾉ ﾂﾅﾃﾞｶﾅｼﾓ  ":RETURN
2210 AD$="ﾌﾙｻﾄｻﾑｸ ｺﾛﾓｳﾂﾅﾘ  ":RETURN
2215 AD$="ﾜｶﾞﾀﾂﾔﾏﾆ ｽﾐｿﾞﾒﾉｿﾃﾞ  ":RETURN
2220 AD$="ﾌﾘﾕｸﾓﾉﾊ ﾜｶﾞﾐﾅﾘｹﾘ  ":RETURN
2225 AD$="ﾋﾄｺｿﾐｴﾈ ｱｷﾊｷﾆｹﾘ  ":RETURN
2230 AD$="ﾐｿｷﾞｿﾞﾅﾂﾉ ｼﾙｼﾅﾘｹﾙ  ":RETURN
2235 AD$="ﾖｦｵﾓﾌﾕｴﾆ ﾓﾉｵﾓﾌﾐﾊ  ":RETURN
2240 AD$="ﾕｴｱﾏﾘｱﾙ ﾑｶｼﾅﾘｹﾘ  ":RETURN
2245 AD$="ｻｼﾓｼﾗｼﾞﾅ ﾓﾕﾙｵﾓﾋｦ  ":RETURN
READY.
```

《写真14―8》下の句

使い方

前50首のプログラムと後50首のプログラムを準備します。できれば，同一テープに両プログラムを記録しておくと便利です。

テープレコーダーにテープを入れて，SHIFT RUNとタイプするとロード終了とともに直ちに実行に入り，**写真14―11**が画面にディスプレイされます。

その後は，前50首の中から乱数によって選ばれた5・7・5およびそれに続いて7・7の句が順にディスプレイされます。**写真14―13**がその実行例です。

前50首が終わると，**写真14―12**に示す文字が画面に現われて，テープレコーダーが自動的に回り（但し，PLAYボタンはすでに押してあるものとして）始め後の50首がロードされます。

後のプログラムは，前のプログラムの中でLOADされた場合にはロード終了とともに実行に移ります。

```
LIST20-80
 20 REM "** アトノ 50 シュ **"
 30 PRINT""
 40 PRINT"ヒャクニン イッシュ"
 50 PRINT"アトノ 50 シュ"
 60 TM=TI
 70 IF (TI-TM)<500 THEN 70
 80 PRINT""
 429 H0=H(S)
 430 ON H0 GOTO 1500,1505,1510,1515,1520,1525,1530,1535,1540,1545
 432 H1=H(S)-10
 433 ON H1 GOTO 1550,1555,1560,1565,1570,1575,1580,1585,1590,1595
 434 H2=H(S)-20
 435 ON H2 GOTO 1600,1605,1610,1615,1620,1625,1630,1635,1640,1645
 436 H3=H(S)-30
 437 ON H3 GOTO 1650,1655,1660,1665,1670,1675,1680,1685,1690,1695
 438 H4=H(S)-40
 439 ON H4 GOTO 1700,1705,1710,1715,1720,1725,1730,1735,1740,1745
 450 GOTO 310
 460 PRINT""
 470 PRINT"アトノ 50 シュ ノ オワリ"
 1500 A$="アケヌレバ クルルモノトハシリナガラ   ":GOTO 500
 1505 A$="ナゲキツツ ヒトリヌルヨノ アクルマハ   ":GOTO 500
 1510 A$="ワスレジノ ユクスエマデハ カタケレバ   ":GOTO 500
 1515 A$="タキノネハ タエテヒサシク ナリヌレド   ":GOTO 500
 1520 A$="アラザラム コノヨノホカノ オモイデニ   ":GOTO 500
 1525 A$="メグリアイテ ミシヤソレトモ ワカヌマニ   ":GOTO 500
 1530 A$="アリマヤマ イナノササハラ カゼフケバ   ":GOTO 500
 1535 A$="ヤスラハデ ネナマシモノヲ サヨフケテ   ":GOTO 500
 1540 A$="オオエヤマ イクノノミチノ トオケレバ   ":GOTO 500
 1545 A$="イニシエノ ナラノミヤコノ ヤエザクラ   ":GOTO 500
 1550 A$="ヨヲコメテ トリノソラネハ ハカルトモ   ":GOTO 500
 1555 A$="イマハタダ オモヒタエナム トバカリヲ   ":GOTO 500
 1560 A$="アサボラケ ウジノカワギリ タエダエニ   ":GOTO 500
 1565 A$="ウラミワビ ホサヌソデダニ アルモノヲ   ":GOTO 500
 1570 A$="モロトモニ アハレトオモヘ ヤマザクラ   ":GOTO 500
 1575 A$="ハルノヨノ ユメバカリナル タマクラニ   ":GOTO 500
 1580 A$="ココロニモ アラデウキヨニ ナガラヘバ   ":GOTO 500
 1585 A$="アラシフク ミムロノヤマノ モミヂバハ   ":GOTO 500
 1590 A$="サビシサニ ヤドヲタチイデテ ナガムレバ ":GOTO 500
 1595 A$="ユウサレバ カドタノイナハ オトヅレテ   ":GOTO 500
 1600 A$="オトニキク タカシノハマノ アダナミハ   ":GOTO 500
 1605 A$="タカサゴノ オノエノサクラ サキニケリ   ":GOTO 500
 1610 A$="ウカリケル ヒトヲハツセノ ヤマオロシ   ":GOTO 500
 1615 A$="ナニオハバ アイサカヤマノ サネカヅラ   ":GOTO 500
 1620 A$="ヲグラヤマ ミネノモミジバ ココロアラバ  ":GOTO 500
 1625 A$="セヲハヤミ イワニセカルル タキガワノ   ":GOTO 500
 1630 A$="アワジシマ カヨフチドリノ ナクコエニ   ":GOTO 500
 1635 A$="アキカゼニ タナビククモノ タエマヨリ   ":GOTO 500
 1640 A$="ナガカラム ココロモシラズ クロカミノ   ":GOTO 500
 1645 A$="アサボラケ アリアケノツキト ミルマデニ  ":GOTO 500
 1650 A$="ホトトギス ナキツルカタヲ ナガムレバ   ":GOTO 500
 1655 A$="オモイワビ サテモイノチハ アルモノヲ   ":GOTO 500
 1660 A$="ヨノナカヨ ミチコソナケレ オモヒイル   ":GOTO 500
 1665 A$="ナガラヘバ マタコノコロヤ シノバレム   ":GOTO 500
 1670 A$="ヨモスガラ モノオモフコロハ アケヤラム  ":GOTO 500
 1675 A$="ナゲケトテ 月ヤハモノヲ オモハスル   ":GOTO 500
 1680 A$="ムラサメノ ツユモマダヒヌ マキノハニ   ":GOTO 500
 1685 A$="ナニハヘノ アシノカリネノ ヒトヨユエ   ":GOTO 500
 1690 A$="タマノオヨ タエナバタエネ ナガラエバ   ":GOTO 500
 1695 A$="ミセバヤナ ヲジマノアマノ ソデダニモ  ":GOTO 500
 1700 A$="キリギリス ナクヤシモヨノ サムシロニ   ":GOTO 500
 1705 A$="ワガソデハ シオヒニミエヌ オキノイシノ  ":GOTO 500
 1710 A$="アフコトノ タエテシナクハ ナカナカニ   ":GOTO 500
 1715 A$="ミヨシノノ ヤマノアキカゼ サヨフケテ   ":GOTO 500
READY.
```

《写真14—9》後50首のプログラム（その1）

```
 1720 A$="オホケナク ウキヨノタミニ オホフカナ   ":GOTO 500
 1725 A$="ハナサソフ アラシノニワノ ユキナラデ   ":GOTO 500
 1730 A$="コヌヒトヲ マツホノウラノ ユウナギニ   ":GOTO 500
 1735 A$="カゼソヨグ ナラノオガワノ ユフグレハ ":GOTO 500
 1740 A$="ヒトモオシ ヒトモウラメシ アヂキナク   ":GOTO 500
 1745 A$="モモシキヤ フルキノキハノ シノブニモ   ":GOTO 500
 2000 AD$="トヒトニハツゲヨ アマノツリフネ  ":RETURN
 2005 AD$="コロモホスチフ アマノカグヤマ  ":RETURN
 2010 AD$="オトメノスガタ シバシトドメム  ":RETURN
 2015 AD$="コヒゾツモリテ フチトナリヌル  ":RETURN
 2020 AD$="ミダレソメニシ ワレナラナクニ  ":RETURN
 2025 AD$="ワガコロモデニ ユキワフリツツ  ":RETURN
 2030 AD$="マツトシキカバ イマカエリコム  ":RETURN
 2035 AD$="カラクレナイニ ミズククルトハ  ":RETURN
 2040 AD$="ユメノカヨイジ ヒトメヨクラム  ":RETURN
 2045 AD$="アワデコノヨオ スグシテヨトヤ  ":RETURN
 2050 AD$="ワガコロモデハ ツユニヌレツツ  ":RETURN
 2055 AD$="ナガナガシヨヲ ヒトリカモネム  ":RETURN
 2060 AD$="フジノタカネニ ユキハフリツツ  ":RETURN
 2065 AD$="コエキクトキゾ アキワカナシキ  ":RETURN
 2070 AD$="シロキヲミレバ ヨゾフケニケル  ":RETURN
 2075 AD$="ヨヲウジヤマト ヒトハイフナリ  ":RETURN
 2080 AD$="ワガミヨニフル ナガメセシマニ  ":RETURN
 2085 AD$="シルモシラヌモ アフサカノセキ  ":RETURN
 2090 AD$="ミヲツクシテモ アハムトゾオモフ  ":RETURN
 2095 AD$="アリアケノ月ヲ マチイデツルカナ  ":RETURN
 2100 AD$="ムベヤマカゼヲ アラシトイフラム  ":RETURN
 2105 AD$="ワガミヒトツノ アキニハアラネド  ":RETURN
 2110 AD$="モミジノニシキ カミノマニマニ  ":RETURN
 2115 AD$="ヒトニシラレデ クルヨシモガナ  ":RETURN
 2120 AD$="イマヒトタビノ ミユキマタナム  ":RETURN
 2125 AD$="イツミキトテカ コイシカルラム  ":RETURN
 2130 AD$="ヒトメモクサモ カレヌトオモヘバ  ":RETURN
 2135 AD$="オキマドハセル シラギクノハナ  ":RETURN
 2140 AD$="アカツキバカリ ウキモノハナシ  ":RETURN
 2145 AD$="ヨシノノサトニ フレルシラユキ  ":RETURN
 2150 AD$="ナガレモアエヌ モミジナリケリ  ":RETURN
 2155 AD$="シズココロナク ハナノチルラム  ":RETURN
 2160 AD$="マツモムカシノ トモナラナクニ  ":RETURN
 2165 AD$="ハナゾムカシノ カニニホイケル  ":RETURN
 2170 AD$="クモノイズコニ 月ヤドルラム  ":RETURN
 2175 AD$="ツラヌキトメヌ タマゾチリケル  ":RETURN
 2180 AD$="ヒトノイノチノ オシクモアルカナ  ":RETURN
 2185 AD$="アマリテナドカ ヒトノコイシキ  ":RETURN
 2190 AD$="モノヤオモフト ヒトノトフマデ  ":RETURN
 2195 AD$="ヒトシレズコソ オモヒソメシカ  ":RETURN
 2200 AD$="スエノマツヤマ ナミコサジトハ  ":RETURN
 2205 AD$="ムカシハモノヲ オモハザリケリ  ":RETURN
 2210 AD$="ヒトヲモミヲモ ウラミザラマシ  ":RETURN
 2215 AD$="ミノイタズラニ ナリヌベキカナ  ":RETURN
 2220 AD$="ユクエモシラヌ コイノミチカナ  ":RETURN
 2225 AD$="ヒトコソミエネ アキハキニケリ  ":RETURN
 2230 AD$="クダケテモノヲ オモフコロカナ  ":RETURN
 2235 AD$="ヒルハキエツツ モノヲコソオモヘ  ":RETURN
 2240 AD$="ナガクモガナト オモヒケルカナ  ":RETURN
 2245 AD$="サシモシラジナ モユルオモヒヲ  ":RETURN
READY.
```

《写真14—10》後50首のプログラム（その2）

```
ヒャクニン イッシュ

マエ 50 シュ
```

《写真14—11》実行

実行例は，**写真14—13**のとおりです。

最後には，**写真14—14**が画面に現われ百首全部のディスプレイが終了します。

なお，REM文にカナを使用するときは，〝 〞で囲む必要がありますので注意してください。

マエ 50 シュ ノ オワリ

アト 50 シュ ワ TAPE カラ LOAD サレマス

《写真14―12》後50首のロード

《写真14―14》実行終了

《写真14―13》実行例

15 ストリング変数の利用法

占い数あてられゲーム

MZ-80K

福島憲一

占い数あてられゲーム

コンピュータの考えた（？）数を人間が当てるというのはマンネリなので，ここでは人間の考えた数をコンピュータが念力でピタリと当てるというのを2連発で作ってみました。RUNすると好きな奇数を入力する様に言ってくるので，例えば5と入力すると今度は5個の数を考える様言って来るので紙にでも好きな数を5個書いて下さい。数は小数でも負数でもかまいません。以下コンピュータの質問に答えていくと，あなたの考えた数全部をピタリと言い当てます。

ただし，コンピュータの質問に嘘を言ってはいけません。ただ答えるのでは面白くないので，どっかの占い師風に変な事を口走ってから答える様にしました。言葉は乱数で組合せているので運が良ければ全裸の百恵チャンが出て来ます。その代り赤いリボン付けた広岡監督も出て来ます。キモチ　ワリー！　その後は説明を要しないでしょう。僕のマイコンは念力を持ってると友達に見せては

《写真15―1》五つの数字を念力で当てる

《写真15―2》思った数字をピタリと念力で！

《写真15―3》マイコンがあらぬことばを……

```
ｵｺﾀｴ ｲﾀｼ ﾏｽ

ｷ ｶﾞ ｸﾙｯﾀ ｺﾏﾜﾘ ｸﾝ ｶﾞ
ｺﾞｷﾌﾞﾘ ﾎｲﾎｲ ﾆ ｶｶｯﾃ ﾍｿ ｦ ﾅﾃﾞﾃ ｲﾙ ﾉｶﾞ ﾐｴﾏｽ

        ｽﾅﾜﾁ
 ｱﾅﾀ ﾉ ｶﾝｶﾞｴﾀ ｶｽﾞ ﾊ
0          18          8          0
12
ｲｶｶﾞ ﾃﾞｽｶ
ﾂｷﾞ ﾆ ｲｷﾏｽ ﾉﾃﾞ KEY ｦ ﾄﾞﾚｶ ｳｯﾃ ｸﾀﾞｻｲ
```

```
ｵｺﾀｴ ｲﾀｼ ﾏｽ

ﾊﾗ ｦ ﾍﾗｼﾀ ｳﾙﾄﾗ ﾏﾝ ｶﾞ
ｺﾞﾊﾝ ﾀﾍﾞﾅｶﾞﾗ ｺｳｿｸ ﾄﾞｳﾛ ｦ ｱﾙｲﾃｲﾙ ﾉｶﾞ ﾐｴﾏｽ

        ｽﾅﾜﾁ
 ｱﾅﾀ ﾉ ｶﾝｶﾞｴﾀ ｶｽﾞ ﾊ
26          -5          17          26
-14
ｲｶｶﾞ ﾃﾞｽｶ
ﾂｷﾞ ﾆ ｲｷﾏｽ ﾉﾃﾞ KEY ｦ ﾄﾞﾚｶ ｳｯﾃ ｸﾀﾞｻｲ
```

《写真15—4》あてられた数字は

```
 10 PRINT"↓→→ ﾈﾝﾘｷ ｿﾉ 1"
 20 DIM NANNO$(7),DAREGA$(8),DOSITE$(8),
SURU$(8)
 30 NA$(0)="ｱｶｲ ﾌｸ ｷﾀ "
 32 NA$(1)="ｻﾐｼｿｳ ﾅ "
 34 NA$(2)="ｵｼｯｺ ｶﾞﾏﾝ ｼﾃﾙ "
 36 NA$(3)="ｷ ｶﾞ ｸﾙｯﾀ "
 38 NA$(4)="ｽｹﾍﾞｰ ﾅ "
 40 NA$(5)="ﾘﾎﾞﾝ ｦ ﾂｹﾀ "
 42 NA$(6)="ﾊﾗ ｦ ﾍﾗｼﾀ "
 44 NA$(7)="ﾏﾙﾊﾀﾞｶ ﾉ "
 50 DA$(0)="ｱﾅﾀ ｶﾞ "
 52 DA$(1)="ｿｳﾘ ﾀﾞｲｼﾞﾝ ｶﾞ "
 54 DA$(2)="ﾓﾓｴ ﾁｬﾝ ｶﾞ "
 56 DA$(3)="ｱﾒｰﾊﾞｰ ｶﾞ "
 58 DA$(4)="ｺﾏﾜﾘ ｸﾝ ｶﾞ "
 60 DA$(5)="ﾓﾝｼﾛﾁｮｳ ｶﾞ "
 62 DA$(6)="ﾀｶﾐﾔﾏ ｶﾞ "
 64 DA$(7)="ﾋﾛｵｶ ｶﾝﾄｸ ｶﾞ "
 66 DA$(8)="ｳﾙﾄﾗ ﾏﾝ ｶﾞ "
 70 DO$(0)="ｸｼﾞﾗ ｶｶｴﾃ "
 72 DO$(1)="ｺﾞｷﾌﾞﾘ ﾎｲﾎｲ ﾆ ｶｶｯﾃ "
 74 DO$(2)="ｱﾒ ﾆ ﾇﾚﾃ "
 76 DO$(3)="ｽｷﾅ ﾋﾄ ﾆ ﾌﾗﾚﾃ "
 78 DO$(4)="ｺﾀﾂ ｶｶｴﾃ "
 80 DO$(5)="ｺﾞﾊﾝ ﾀﾍﾞﾅｶﾞﾗ "
 82 DO$(6)="ﾖｯﾊﾟﾗｯﾃ "
 84 DO$(7)="ﾖﾀﾞﾚ ﾀﾗｼﾃ "
 86 DO$(8)="ﾎﾟﾙｼｪ ﾆ ﾉｯﾃ "
 90 SU$(0)="ﾍｿ ｦ ﾅﾃﾞﾃ ｲﾙ "
 92 SU$(1)="ﾄﾝﾃﾞ ｲｸ "
 94 SU$(2)="ｵﾄﾞｯﾃ ｲﾙ "
 96 SU$(3)="ﾅｷｻｹﾝﾃﾞ ｲﾙ "
 98 SU$(4)="ｶﾗｵｹ ﾃﾞ ｳﾀｯﾃｲﾙ "
 100 SU$(5)="ｻｶﾀﾞﾁ ｼﾃｲﾙ "
 102 SU$(6)="ｺｳｿｸ ﾄﾞｳﾛ ｦ ｱﾙｲﾃｲﾙ "
 104 SU$(7)="ｷﾞｮｳｻﾞ ｦ ﾀﾍﾞﾃｲﾙ "
 106 SU$(8)="ﾄｲﾚ ﾆ ﾄﾋﾞｺﾑ "
 110 PRINT"↓→→ｽｷﾅ ｷｽｳ ｦ ﾆｭｳﾘｮｸ ｼﾃ ｸﾀﾞｻｲ"
:MUSIC"C1‾C1"
 120 INPUT K:IFK/2=INT(K/2)THEN PRINT"ｷｽ
ｳ ﾀﾞｹ!":GOTO120
 130 DIM A(K),B(K)
 140 PRINT"↓→→";K;"ｺ ﾉ ｶｽﾞ ｦ ｶﾝｶﾞｴﾃ ｸﾀﾞ
ｻｲ":MUSIC"E2C2"
 150 PRINT"↓ﾄﾞﾝﾅ ｶｽﾞ ﾃﾞﾓ ｶﾏｲﾏｾﾝ"
 160 PRINT"↓→ｶﾝｶﾞｴﾀﾗ ﾜﾀｼ ﾉ ｼﾂﾓﾝ ﾆ ｺﾀｴﾃ ｸ
ﾀﾞｻｲ":PRINT
 170 FORI=1TOK:J=I+1
 180 IFI=KTHENJ=1
 190 PRINTI;" ﾊﾞﾝﾒ ﾉ ｶｽﾞ ﾄ";J;" ﾊﾞﾝﾒ ﾉ ｶ
ｽﾞ ｦ":PRINT
 200 PRINT"  ﾀｽﾄ\ｲｸﾂ ﾆ ﾅﾘﾏｽｶ?":MUSIC"B1"
 210 INPUT A(I)
 220 NEXT I
 230 A=0:B=0
 240 FOR I=1TO K
 250 IF I/2=INT(I/2)THEN B=B+A(I):GOTO27
0
 260 A=A+A(I)
 270 NEXT I
 280 B(1)=(A-B)/2
 290 FOR I=2 TO K
 300 B(I)=A(I-1)-B(I-1):NEXT I
 310 GOSUB 700
 320 FORI=1TOK:PRINTB(I),:MUSIC"‾G2":NEX
T I
 330 PRINT:PRINT:PRINT"ｲｶｶﾞ ﾃﾞｽｶ  "
 340 PRINT"↓ﾂｷﾞ ﾆ ｲｷﾏｽ ﾉﾃﾞ KEY ｦ ﾄﾞﾚｶ ｳｯ
ﾃ ｸﾀﾞｻｲ"
 350 GET Z$:IF Z$=""THEN350
 400 PRINT"↓→→ ﾈﾝﾘｷ ｿﾉ 2":DIM C(2)
 410 PRINT"↓→→1 ｲｼﾞｮｳ ﾃﾞ 59 ｲｶ ﾉ ｽｷﾅ ｶｽﾞ
ｦ":PRINT"↓ ｶﾝｶﾞｴﾃ ｸﾀﾞｻｲ":PRINT
 420 MUSIC" E2R2_G"
 430 PRINT"↓→ｶﾝｶﾞｴﾀﾗ ﾜﾀｼ ﾉ ｼﾂﾓﾝ ﾆ ｺﾀｴﾃ ｸ
ﾀﾞｻｲ":PRINT
 440 FOR I=0TO2:MUSIC"C0FA"
 450 PRINT"↓→→ｿﾉ ｶｽﾞ ｦ";I+3;" ﾃﾞ ﾜｯﾀ ﾄｷﾉ
ｱﾏﾘ ﾊ?"
 460 INPUT C(I):PRINT:NEXT I
 470 M=INT(C(0))*40+INT(C(1))*45+INT(C(2
))*36
 480 N=M-INT(M/60)*60
 490 GOSUB700:GOSUB810
 500 PRINT:PRINTN;" ﾃﾞｽﾈ"
 510 PRINT"↓ﾜﾀｼ ﾆ ﾁｮｳﾉｳﾘｮｸ ｶﾞ ｱﾙﾉｶﾞ ﾜｶﾘﾏ
ｼﾀｶ?"
 520 PRINT"↓ﾀｶｶﾞ ﾏｲｺﾝ ﾄ ﾊﾞｶ ﾆ ｽﾙﾄ":PRINT
:PRINT"    ﾊﾞﾁ ｶﾞ ｱﾀﾘﾏｽﾖ!!":PRINT
 530 END
 700 PRINT"↓→→→ｵｺﾀｴ ｲﾀｼ ﾏｽ":PRINT:PRINT
:PRINT:GOSUB810
 710 R=INT(RND(1)*8):PRINTNA$(R);:GOSUB8
10
 720 R=INT(RND(1)*9):PRINTDA$(R):PRINT:G
OSUB 810
 730 R=INT(RND(1)*9):PRINTDO$(R);:GOSUB
810
 740 R=INT(RND(1)*9):PRINTSU$(R);"ﾉｶﾞ ﾐｴ
ﾏｽ":GOSUB 810
 750 PRINT:PRINT:PRINT"→→→→→→→→→ｽﾅﾜﾁ":PR
INT:GOSUB810
 760 PRINT"   ｱﾅﾀ ﾉ ｶﾝｶﾞｴﾀ ｶｽﾞ ﾊ":PRINT
 770 GOSUB 810:RETURN
 800 MUSIC"D1F":FOR T=0TO4000:NEXT:RETUR
N
 810 MUSIC"D1F":FOR T=0TO600:NEXT:RETURN
```

占い数あてられゲーム

いかが？　きっとウケると思います。

なおSP—5010高速BASICを使っている方はGOSUB　810を800に代えて下さい。

16

効果音を活かした

ブロックくずし

TK80BSLII

尾島辰彦

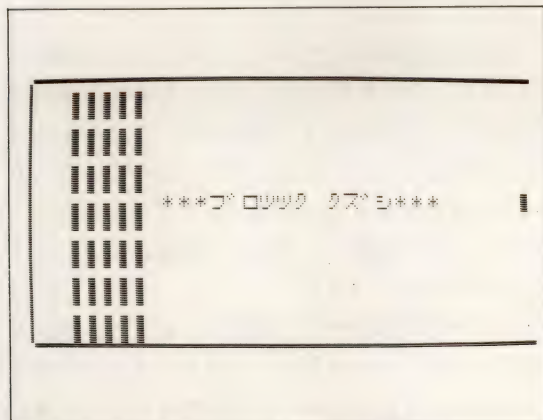
ゲームのやり方

プログラムをロードしてRUNさせると，タイトル，コート，ブロック，パドルが画面に出ます。

ブロックは左側に配置して横型にしてあります。

タイトルが消えると，タマがサーブされます。Uキーと Hキーを使ってパドルを上下に動かし，うまくタマを受けると，はね返り，ブロックに当るとブロックを消してはね返ります。

Uキー，Hキー以外のキーを押すと，パドルは止まります。パドルはタマが動いている間だけ動かすことができます。

タマを受けそこなうと，そこでタワは消え，持ちダマの残りの数の表示が出て再サーブとなります。

ブロックの数が15より少なくなるとタマはだんだん速く動くようになっています。

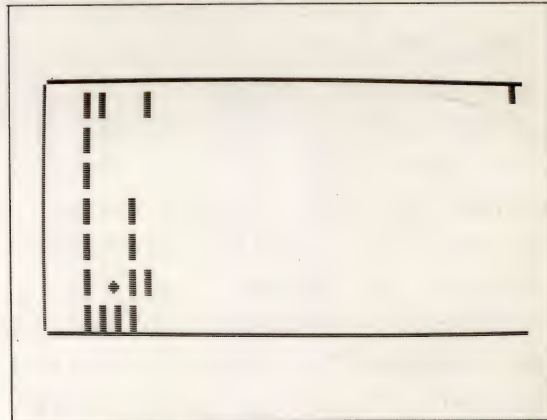
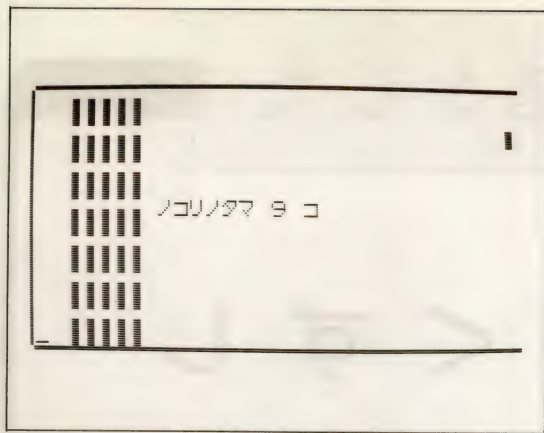
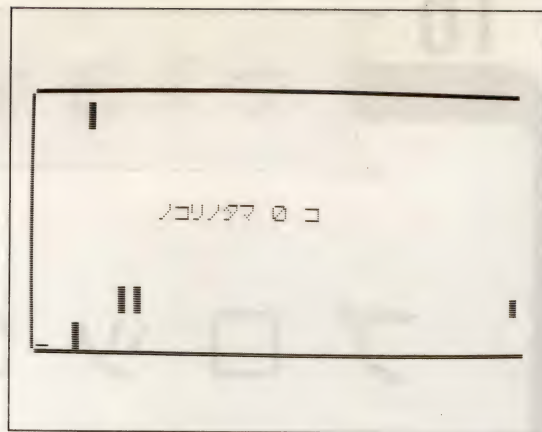
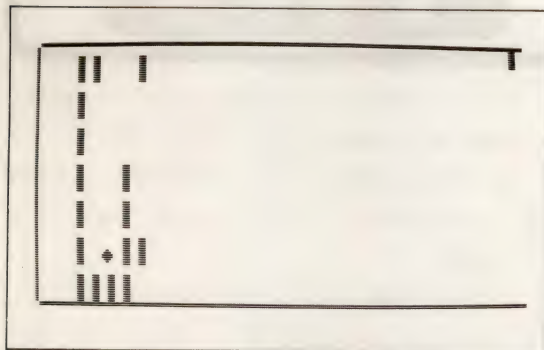
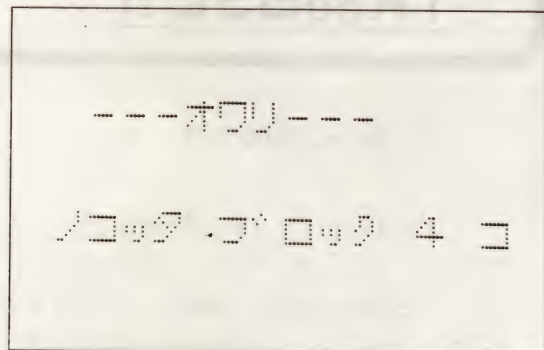
持ちダマは10個で，うまく全部のブロックを消すことができたら，メッセージが出て再ゲームとなります。

ブロックが残っているのにタマがなくなってしまったら，メッセージと残りのブロック数を表示してストップします。

プログラムの説明

このプログラムは，大きく三つの部分に分けられます。10～310は初期状態のセット，400，600，800，1000の四つのルーチンが主要部，1400以降がサブルーチン，処理ルーチン群です。

1300～1330は470，670，870，1070のフルスペル

《写真16—1》ゲームスタート

《写真16—2》ブロックをくずす

《写真16—3》持ちダマの表示

《写真16—4》ブロック数が15より少なくなると玉の速度が増す

《写真16—5》全部消すと再ゲーム

《写真16—6》残りがあるとゲーム終了

換算で，80字をこえるため，二つに分けた後半です。

IF文なので，となりに入れるわけにはいかないので，やむを得ずこのようにしました。

タマの移動は，ビデオRAMのアドレスAにタマのキャラクタコードCAHを書込み，そのアドレスに，+33，−31，+31，−33を加えることにより行ないます。

タマを1マス進めるたびに元の場所には空白（20H）を書込みます。このアドレスをBとします。BはAの後を常に1マス遅れてついて行くようにしてあります。

タマを動かす方向は，右下，右上，左下，左上の4方向があり，動かす方向に対応して四つのルーチンがあります。これをそれぞれ右下ルーチン，左上ルーチンなどと呼びます。

この四つのルーチンは相互に複雑に連絡して閉ループ系を形成しています。ループの入口は400または，600で，どちらから入るかは290のCで決まります。

タマをパドルで受けそこなったときのみ，このループから抜け出します。ブロックはA9H，83Hの二つのキャラクターを組み合わせて作ってあります。したがってタマがどちらに当っても残りの半分を同時に消すようにしてあります。

プログラムの流れ

10〜210で画面をセットし，250でタマの数Vを10，ブロックの数Wを35にセットします。260はタマは受けそこねたとき，1700のルーチンからもどって来るところで，ここでブロックがなくなったかどうか判定します。270では持ちダマがなくなったかどうかを判定します。その結果により対応するルーチンへとびます。

次に280〜310でサーブを行い，右下ルーチンか右上ルーチンへ行きます。これからあとはタマをミスするまで四つのルーチンの間をいったり来たりするわけです。

ブロックくずしゲームプログラム

```
10 CLEAR
20 CURSOR 10,8: PRINT "***ﾌﾞﾛｯｸ ｸｽﾞｼ***"
30 FOR X=32 TO 2 STEP -1
40 CURSOR X,1: PICTURE 95: NEXT X
50 FOR Y=2 TO 15
60 CURSOR 1,Y: PICTURE 87: NEXT Y
70 FOR X=2 TO 32
80 CURSOR X,16: PICTURE 92: NEXT X
85 LET M=32511
90 POKE M,82H
100 FOR X=4 TO A
110 FOR Y=2 TO 14 STEP 2
120 CURSOR X,Y: PICTURE A9
130 NEXT Y: NEXT X
140 FOR X=4 TO 8
150 FOR Y=3 TO 15 STEP 2
160 CURSOR X,Y: PICTURE 83
170 NEXT Y: NEXT X
200 GOSUB 1600
210 GOSUB 1400
250 LET V=10,W=35
260 IF W=0 THEN 2100
270 IF V=0 THEN 2000
280 RANDOMIZE : LET A=32361+INT(RND(9))*32
290 LET C=INT(RND(2))+1
300 GOSUB 1600
310 ON C GOTO 400,600
400 REM ※ﾐｷﾞｼﾀ ﾙｰﾁﾝ
410 GOSUB 1500
420 IF PEEK(7DFCH)=85 THEN GOSUB 3000
430 IF PEEK(7DFCH)=72 THEN GOSUB 3100
440 LET B=A
450 LET A=A+33
460 LET D=A-32,E=A+32
470 IF PEEK(A)=131 THEN IF A>32705 THEN 1300
480 IF PEEK(A)=131 THEN POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 800
490 IF PEEK(A)=169 THEN POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 800
500 POKE A,CAH: POKE B,20H
510 IF A=32734 THEN IF M=A+1 THEN 1000
520 IF A=32734 THEN 1700
530 IF A>32705 THEN 600
540 IF (A+2)/32-INT((A+2)/32)>0 THEN 400
550 IF M=A+1 THEN 800
560 GOTO 1700
600 REM ※ﾐｷﾞｳｴ ﾙｰﾁﾝ
601 CALL 8306H
602 CALL 8300H
610 GOSUB 1500
620 IF PEEK(7DFCH)=85 THEN GOSUB 3000
630 IF PEEK(7DFCH)=72 THEN GOSUB 3100
640 LET B=A
650 LET A=A-31
660 LET D=A-32,E=A+32
670 IF PEEK(A)=169 THEN IF A<32318 THEN 1310
680 IF PEEK(A)=169 THEN POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 1000
690 IF PEEK(A)=131 THEN POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 1000
700 POKE A,CAH: POKE B,20H
710 IF A=32318 THEN IF M=A+1 THEN 800
720 IF A=32318 THEN 1700
730 IF A<32318 THEN 400
740 IF (A+2)/32-INT((A+2)/32)>0 THEN 600
750 IF M=A+1 THEN 1000
760 GOTO 1700
800 REM ﾋﾀﾞﾘｼﾀ ﾙｰﾁﾝ
801 CALL 8306H
802 CALL 8300H
810 GOSUB 1500
820 IF PEEK(7DFCH)=85 THEN GOSUB 3000
830 IF PEEK(7DFCH)=72 THEN GOSUB 3100
840 LET B=A
850 LET A=A+31
860 LET D=A-32,E=A+32
870 IF PEEK(A)=131 THEN IF A>32705 THEN 1320
880 IF PEEK(A)=131 THEN POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 400
890 IF PEEK(A)=169 THEN POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 400
900 POKE A,CAH: POKE B,20H
910 IF A=32705 THEN 600
920 IF (A-1)/32-INT((A-1)/32)=0 THEN 400
930 IF A>32705 THEN 1000
940 GOTO 800
1000 REM ﾋﾀﾞﾘｳｴ ﾙｰﾁﾝ
1001 CALL 8306H
1002 CALL 8300H
1010 GOSUB 1500
1020 IF PEEK(7DFCH)=85 THEN GOSUB 3000
1030 IF PEEK(7DFCH)=72 THEN GOSUB 3100
1040 LET B=A
1050 LET A=A-33
1060 LET D=A-32,E=A+32
1070 IF PEEK(A)=169 THEN IF A<32318 THEN 1330
1080 IF PEEK(A)=169 THEN POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 600
1090 IF PEEK(A)=131 THEN POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 600
1100 POKE A,CAH: POKE B,20H
1110 IF A=32289 THEN 400
1120 IF (A-1)/32-INT((A-1)/32)=0 THEN 600
1130 IF A<32318 THEN 800
1140 GOTO 1000
1300 POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 1000
1310 POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 800
1320 POKE D,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 600
1330 POKE E,20H: POKE A,CAH: POKE B,20H: LET W=W-1: GOTO 400
1400 FOR X=10 TO 31
1410 CURSOR X,8: PICTURE 20: NEXT X
1420 RETURN
1500 IF W>=15 THEN LET P=15
1510 IF W<15 THEN LET P=W
1520 FOR Q=1 TO P: NEXT Q
1530 RETURN
1600 FOR J=1 TO 500
1610 NEXT J
1620 RETURN
1700 GOSUB 1500
1710 POKE A,20H
1720 LET V=V-1
1730 CURSOR 10,8: PRINT "ﾉｺﾘﾉﾀﾏ";V;"ｺ"
1740 CURSOR 2,15
1750 POKE 7F00H,87H
1760 GOSUB 1600
1770 GOSUB 1400
1780 GOTO 260
2000 CLEAR
2010 CURSOR 11,5: PRINT "---ｵﾜﾘ---"
2020 CURSOR 10,8: PRINT "ﾉｺｯﾀ ﾌﾞﾛｯｸ";W;"ｺ"
2030 STOP
2100 CLEAR
2110 CURSOR 10,5: PRINT "---ｵﾒﾃﾞﾄｳ---"
2120 CURSOR 10,8: PRINT "ﾓｳｲﾁﾄﾞ ﾃﾞｷﾏｽ"
2130 GOSUB 1600
2140 GOTO 10
3000 IF M=32319 THEN 3050
3010 LET M=M-32
3020 POKE M,82H
3030 LET N=M+32
3040 POKE N,20H
3050 RETURN
3100 IF M=32735 THEN 3150
3110 LET M=M+32
3120 POKE M,82H
3130 LET N=M-32
3140 POKE N,20H
3150 RETURN
```

```
8300 011020        LXI   B,2010H
8303 C30983        JMP   8309H
8306 010510        LXI   B,1005H
8309 F5            PUSH  PSW
830A C5            PUSH  B
830B D5            PUSH  D
830C E5            PUSH  H
830D 2EFF          MVI   L,FFH
830F 61            MOV   H,C
8310 48            MOV   C,B
8311 3E02          MVI   A,02H
8313 D3FA          OUT   FAH
8315 2B            DCX   H
8316 7C            MOV   A,H
8317 A7            ANA   A
8318 CA3183        JZ    8331H
831B 0D            DCR   C
831C C21583        JNZ   8315H
831F 48            MOV   C,B
8320 3E00          MVI   A,00H
8322 D3FA          OUT   FAH
8324 2B            DCX   H
8325 7C            MOV   A,H
8326 A7            ANA   A
8327 CA3183        JZ    8331H
832A 0D            DCR   C
832B C22483        JNZ   8324H
832E C31083        JMP   8310H
8331 E1            POP   H
8332 D1            POP   D
8333 C1            POP   B
8334 F1            POP   PSW
8335 C9            RET
```

右下ルーチンの説明

四つのルーチンは大体同じ構造なので，右下ルーチンを代表として説明します。

このルーチンに処理がどんどん来ると，まず1500のタイム・サブルーチンへ行きます。これはタマを動かす速さをコントロールするところです。

次に420，430でパドルを動かすキー，U，Hが押されているか判定します。もし押されていたら対応するサブルーチン3000または，3100へとび，パドルを上か下へ1マス進めてリターンします。

次に450でタマのアドレスを1マス（＋33）進めます。この段階ではタマは進めません。次にそのアドレスにブロックがあるかどうかの判定をしますが，下辺に接して置かれているブロックの場合は，判断条件が重なるため，ここだけは別の処理が必要です。470がこれに相当します。もし下辺に接した後，もと来た方向（左上ルーチン）へとびます。

この判断が終了したら一般の場所のブロックを判定します。ブロックは2種のパターンで作ってありますから，どちらに当ってもわかるように二つのIF文480，490で判定します。

もし，ブロックであれば，そこへタマを進め，残り半分も消してはね返る方向の左下ルーチンへとびます。

もし，ブロックがない場合は，450で進めたアドレスに500でタマを進めます。そして，その場所の判断を行ないます。

まず，右下スミの判断を行います。ここも判断条件が重なるところなので，先にやる必要があります。

510は，右下スミでかつパドルが右となりにあるか（すなわち，パドルに当ったか）を判断し，そうであればもと来たコース，左上ルーチンにとびます。

520は右下スミであって，パドルが右となりにない場合ですから，受けそこないです。処理ルーチン1700へとびます。

スミの判断が終わると，530で下辺であるか判定し，そそうであれば，はね返る方向の右上ルーチンへとびます。

最後にパドルが上下する左となりの列に達したかを判定します。この列のアドレスはあと2加えれば32で割り切れるようになるので，この性質を利用して540で判定を行います。

この列に達していなければ，このルーチンの頭にもどり，再び同じことをくり返します。

この列に達すると，540のIF文が通らなくなり，550へ行きます。

ここで右となりにパドルがあるか（パドルに当ったか）を判定し，当っていればはね返る方向の左下ルーチンにとびます。

パドルに当らなかったらミスですから，560から，1700の処理ルーチンへとびます。

右へ行くルーチンでブロックの判定が必要なのは，タマがブロックの後に入り込んで左カベではね返ることがあるからです。

右上ルーチンは上下が反対になるわけで，まったく同じに考えられます。

左下ルーチン，左上ルーチンは，パドルの判定がなく，代わりに左カベの判定があるだけで，基本的にはやはり同じと考えられます。

これらのルーチンを要約すると進行中に起り得るあらゆる状態をIF文で判定し，その結果によって対応するルーチンへとび，どのIF文にもかからなかったら，ルーチンの頭にもどって再びタマを進めるということです。

スミやブロックが辺に接しているところの判断や，処理は，わかりにくいかも知れませんが，ここを手抜きすると，タマがコートの壁をつき抜いて穴をあけてしまいます。

各種ルーチンの説明

1400〜1420：タイトル，残りダマ表示の消去サブルーチン

1500〜1530：短かいタイムサブルーチン，タマの速さをコントロールする。ブロック数で決定

1600〜1620：長いタイムサブルーチン，タイトル，残りダマ表示の保持，サーブの前の時間

1700〜1780：タマをミスしたときの処理，タマを

消して残りダマ数を表示して260へ行く

2000〜2030：持ちダマがなくなったときの処理，残りのブロック数を表示してストップ

2100〜2140：ブロックを全部消したときの処理，メッセージ表示後一番初めにもどる

3000〜3050：パドルを上へ動かすサブルーチン，パドルが上端にあるときそのままリターン

3100〜3150：パドルを上へ動かすサブルーチン，下端にあるときはそのままリターン

変数の用途

A：タマのアドレス

B：タマを1マス移動させた後へ空白（20H）を書込むアドレス

C：サーブの方向を決める

D：
E：｝ブロックにタマが当ったとき残り半分を消すためのアドレス

J：1600長いタイムサブルーチン中の制御変数

M：パドルのアドレス

N：パドルを1マス進めた後へ空白（20H）を書込むためのアドレス

P：
Q：｝1500短かいタイムサブルーチン中で，タマの速さをコントロールする

V：タマの数

W：ブロックの数

X：
Y：｝カーソルの座標

17

マイコンに思考を！

オールマイティオセロ

TK-80BSL II

近藤直之

このオセロゲームには次のような特徴があります。

1. グラフィック機能をフルに発揮し，大きく見易い盤面にしてあります。

2. 人間対人間，人間対コンピュータ，コンピュータ同志の対決を可能にしました。

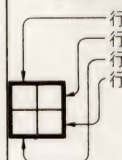
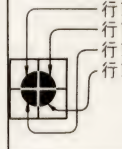
これにより，初心者は，まずコンピュータ同志の対戦を模範手順として見学することにより，オセロの定石をおぼえ，次にコンピュータと対戦して腕をみがき，最後に人間対人間で相手をまかすように練習できます。

3. コンピュータの考える時間を短縮するため，機械語とのリンクも考えられます。そのため盤面の記憶に，TK-80のRAMを8300Hから8363Hまでワーキング・エリアとして用いています。

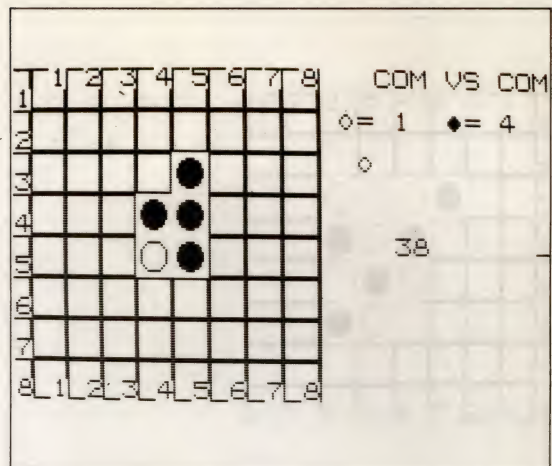
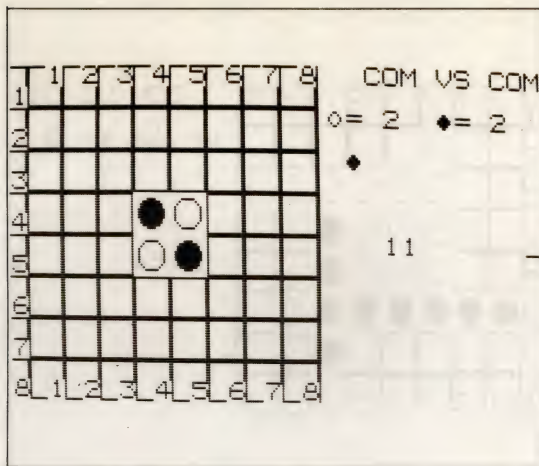
4. 置き場所を入力する際，あわてて〔復改〕キーのみ押してしまったら〝ニュウリョクアヤマリ〞となって盤面をこわしてしまいます。しかし本プログラムでは，その時点の盤面を書き直すルーチンを持っています。

5. なるべくマルチ・ステートメントは用いず，またサブルーチンを多用してデバックの行い易いプログラムとしました。

盤面表示の説明

盤面の記憶にTK-80RAMの8300H～8363Hをワーキングエリアとして用いています。これは，

《第17―1図》盤面表示例

①BSのメモリオーバーフローを防ぐため，②今後機械語で反転可能かどうかの判断プログラムを組み，コンピュータの手のスピードアップを図るため，③人間対人間でも反転可能かどうかを機械語で判断し，パスとか終わりの自動表示を行なわせる。の以上3点を目的としているためです。

これらの記憶内容は，盤の周り（つまり盤の範囲外）に00H＝$(0)_{10}$，駒の置かれていない場所に09H＝$(9)_{10}$，●の置かれている場所に2DH＝

$(45)_{10}$, ○の置かれている場所に31H＝$(49)_{10}$です。

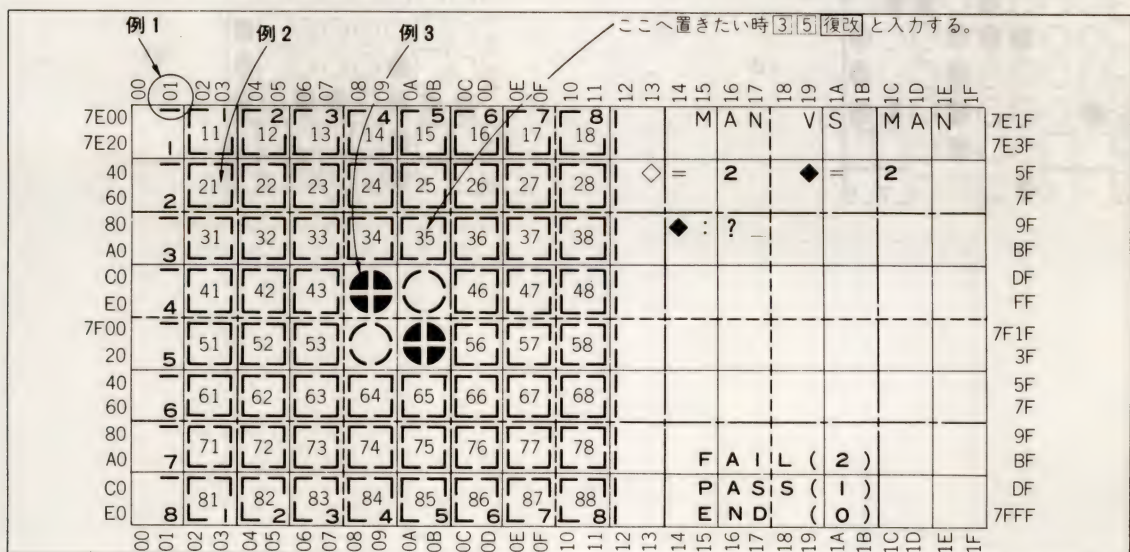
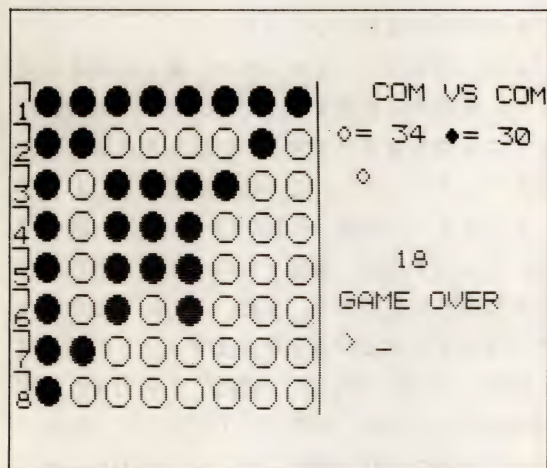
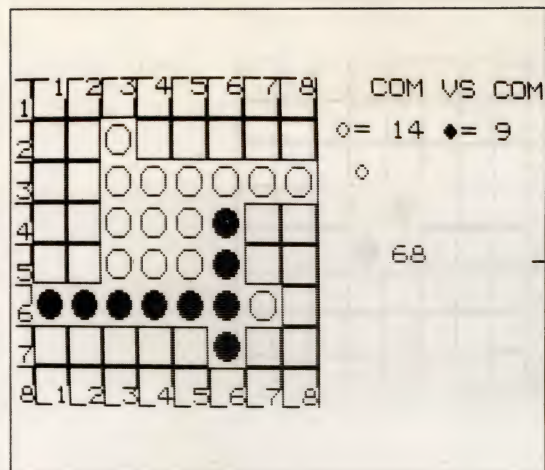
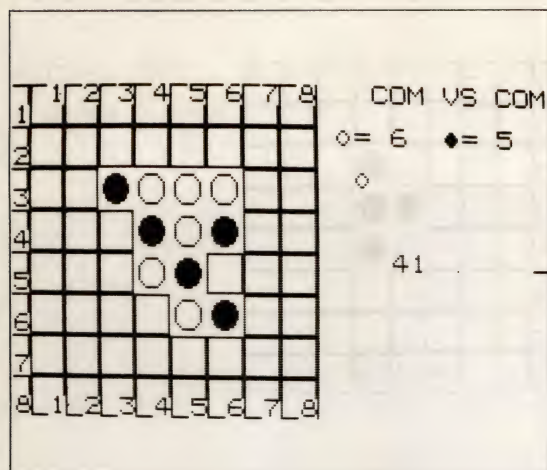
この内容を表示しているのは，行番号1550 PO. X, Yですが，Yの値は，行番号1530の一部，Y＝P. (J)＋I＋149で決ります（**第17―1図**）。

場所入力の方法

画面表示例は**第17―2図**をみて下さい。

たとえば，タテ―3，ヨコ―5の位置に駒を置きたい場合は，〔3〕〔5〕〔復改〕と入力すれば●が3―5に入り4―5の○が●に反転します。

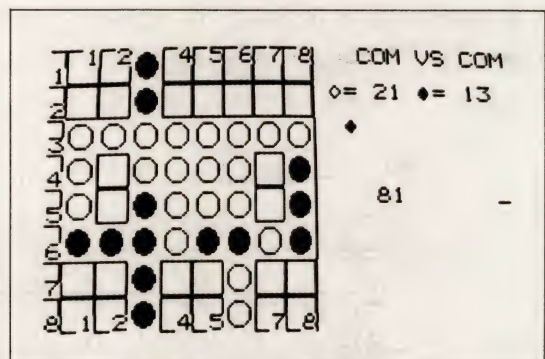
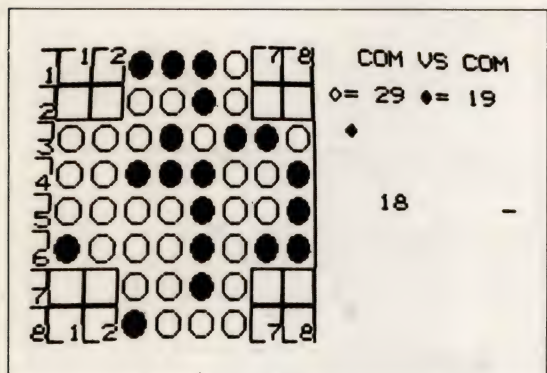
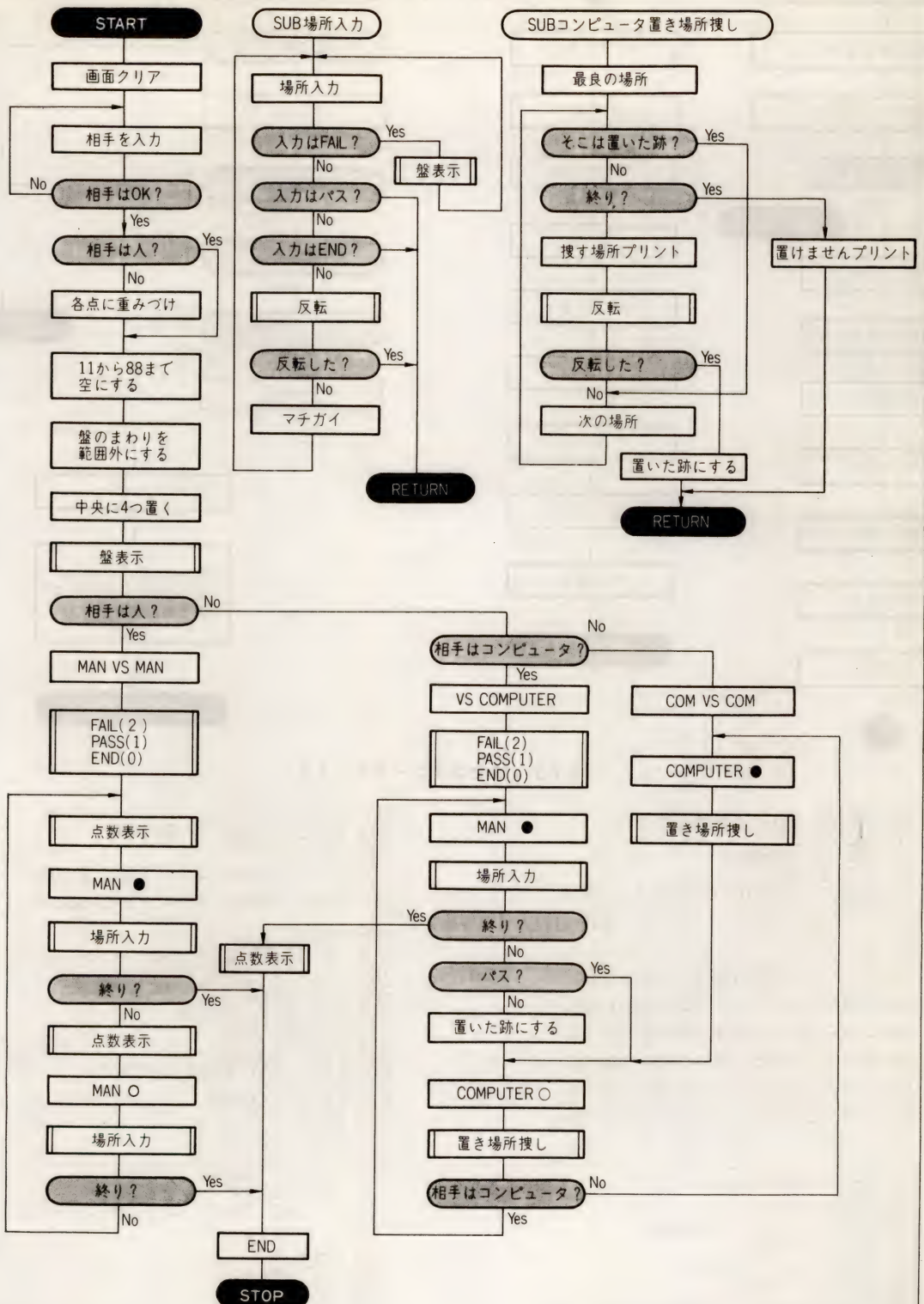
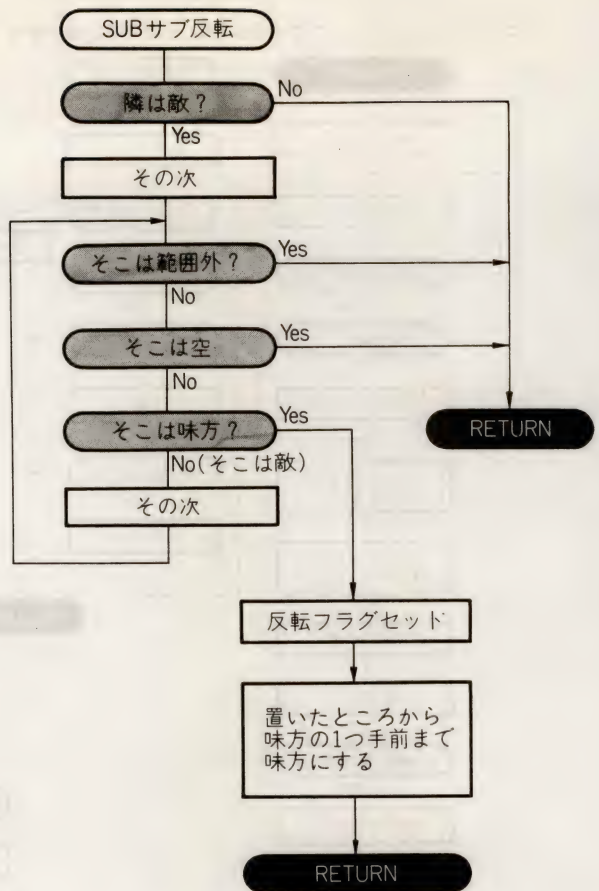
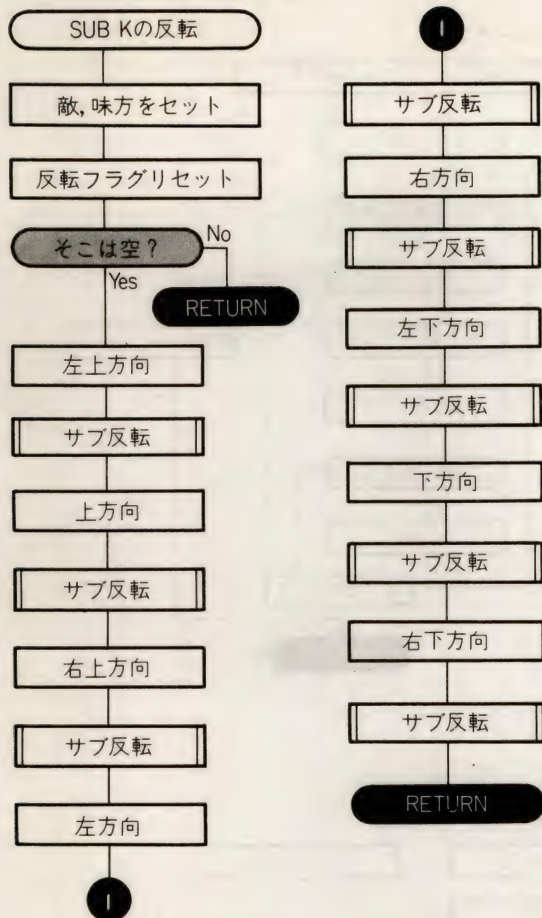
表示例で3―5，4―6，5―3，6―4以外だと相手の駒を反転できませんので，行番号2100で入力された数は，〃マチガイ〃とプリントされて，もう一度入力待ちの状態になります。

また，何も入力しなかったり，数字以外を入力して，復改キーを押すと，インタプリンタの働きで〃**ニュウリョクアヤマリ**〃と表示され入力待ちとなります。しかし，この場合は盤面をこわしてしまいます。この時，FAIL (2) に従って〔2〕〔復改〕と入力すれば，盤面をそのまま書き直し，再度入力待ちとなります。また，どこにも相手をはさむ所がなければパスとなりますが，この場合はPASS (1) に従って，〔1〕〔復改〕と入力すれば相手の番になります。相手も入力できなければゲーム終了なので〔0〕〔復改〕と入力しENDルーチンへ飛びます。

《第17―2図》画面表示例

```
*** OTHELLO ***
PARTNER-MAN( 1 ), COMPUTER( 2 ), NO(3)
?_
```

マイティオセロフローチャートA

SUB Kの反転
敵,味方をセット
反転フラグリセット
そこは空？
No
RETURN
Yes
左上方向
サブ反転
上方向
サブ反転
右上方向
サブ反転
左方向
1

1
サブ反転
右方向
サブ反転
左下方向
サブ反転
下方向
サブ反転
右下方向
サブ反転
RETURN

SUBサブ反転
隣は敵？
No
Yes
その次
そこは範囲外？
Yes
No
そこは空
Yes
No
そこは味方？
Yes
No(そこは敵)
その次
RETURN
反転フラグセット
置いたところから
味方の1つ手前まで
味方にする
RETURN

マイティオセロフローチャートB

```
  10 CLEAR
  20 PRINT "*** OTHELLO ***"
  30 PRINT
  40 LET A=32256
  50 LET B=33536
  60 DIM A(60)
  70 INPUT "PARTNER-MAN(1),COMPUTER(2),NO(3)"C
  80 IF C>0 THEN  IF C<4 THEN 100
  90 GOTO 70
 100 IF C=1 THEN 220
 110 DATA 11,88,18,81,16,61,68,13
 120 DATA 86,38,31,83,15,41,48,51
 130 DATA 85,14,84,58,36,33,66,63
 140 DATA 34,65,35,46,64,43,56,53
 150 DATA 23,62,67,37,26,76,73,32
 160 DATA 24,42,52,47,25,74,75,57
 170 DATA 21,17,71,82,78,87,28,12
 180 DATA 22,72,77,27,100
 190 FOR I=0 TO 60
 200 READ A(I)
 210 NEXT I
 220 FOR X=B+11 TO B+88
 230 POKE X,09H
 240 NEXT X
 250 FOR X=B TO B+10
 260 POKE X,00H
 270 NEXT X
 280 FOR X=B+20 TO B+80 STEP 10
 290 POKE X,00H
 300 NEXT X
 310 FOR X=B+19 TO B+79 STEP 10
 320 POKE X,00H
 330 NEXT X
 340 FOR X=B+89 TO B+99
 350 POKE X,00H
 360 NEXT X
 370 POKE 832CH,2DH: POKE 832DH,31H
 380 POKE 8336H,31H: POKE 8337H,2DH
 390 CLEAR
 400 GOSUB 500
 410 IF C=1 THEN 1000
 420 IF C=2 THEN 1200
 430 GOTO 1400
 500 REM DISPLAY
 510 FOR O=11 TO 88
 520 GOSUB 1500
 530 NEXT O
 540 FOR I=0 TO 7
 550 LET X=A+I*2+3,Y=49+I
 560 POKE X,Y
 570 LET X=X+480
 580 POKE X,Y
 590 NEXT I
 600 FOR I=0 TO 7
 610 FOR J=0 TO 1
 620 LET X=A+I*64+J*32
 630 POKE X,20H
 640 LET X=X+1
 650 IF J=0 THEN  LET Y=159: GOTO 670
 660 LET Y=49+I
 670 POKE X,Y
 680 LET X=X+17
 690 POKE X,88H
 700 NEXT J
 710 NEXT I
 720 RETURN
1000 REM MAN VS MAN
1010 CURSOR 22,1: PRINT "MAN VS MAN";
1020 GOSUB 1600
1030 GOSUB 1700
1040 POKE 7E94H,CAH
1050 GOSUB 2000
1060 IF K=0 THEN 2500
1070 GOSUB 1700
1080 POKE 7E94H,CCH
1090 GOSUB 2000
```

```
 1100 IF K=0 THEN 2500
 1110 GOTO 1030
 1200 REM MAN VS COMPUTER
 1210 CURSOR 22,1: PRINT "VS COM
PUTER";
 1220 GOSUB 1600
 1230 GOSUB 1700
 1240 POKE 7E94H,CAH
 1250 GOSUB 2000
 1260 IF K=0 THEN 2500
 1270 IF K=1 THEN 1310
 1280 FOR I=0 TO 60
 1290 IF A(I)=K THEN  LET A(I)=1
0: GOTO 1310
 1300 NEXT I
 1310 POKE 7E94H,CCH
 1320 GOSUB 1700
 1330 GOSUB 2200
 1340 IF C=2 THEN 1230
 1350 GOTO 1420
 1400 REM COMP VS COMP
 1410 CURSOR 22,1: PRINT "COM VS
COM";
 1420 POKE 7E94H,CAH
 1430 GOSUB 1700
 1440 GOSUB 2200
 1450 GOTO 1310
 1500 REM SUB DISPLAY
 1510 LET G=INT(O/10),H=O-G*10,J
=B+O
 1520 IF H=0 THEN  RETURN
 1530 IF H=9 THEN  RETURN
 1540 FOR I=0 TO 3
 1550 LET X=A+G*64+H*2+I-64,Y=PE
EK(J)+I+149
 1560 IF I>1 THEN  LET X=X+30
 1570 POKE X,Y
 1580 NEXT I
 1590 RETURN
 1600 REM FUNCTION
 1610 POKE 8624H,00H
 1620 CURSOR 22,14: PRINT "FAIL(
2)"
 1630 CURSOR 22,15: PRINT "PASS(
1)"
 1640 CURSOR 22,16: PRINT "END (
0)";
 1650 RETURN
 1700 REM POINT
 1710 CALL 8200H
 1720 LET R=PEEK(82FFH)
 1730 LET S=PEEK(82FEH)
 1740 POKE 8624H,00H
 1750 POKE 7E53H,CCH
 1760 CURSOR 21,3: PRINT "=";S;
 1770 POKE 7E59H,CAH
 1780 CURSOR 27,3: PRINT "=";R;
 1790 IF R+S=64 THEN 2500
 1800 RETURN
 2000 REM INPUT
 2010 POKE 8624H,00H
 2020 CURSOR 22,5: PRINT "      "
;
 2030 CURSOR 22,5: INPUT ":"K
 2040 IF K=2 THEN  GOSUB 500: GO
TO 2010
 2050 IF K=1 THEN  RETURN
 2060 IF K=0 THEN  RETURN
 2070 GOSUB 3000
 2080 IF E=1 THEN 2120
 2090 POKE 8624H,00H
 2100 CURSOR 22,9: PRINT K;"マチガ
イ";
 2110 GOTO 2010
 2120 POKE 8624H,00H
 2130 CURSOR 22,9: PRINT "
   ";
 2140 RETURN
 2200 REM COMPUTER
 2210 POKE 8624H,00H
 2220 CURSOR 22,5: PRINT "
";
 2230 LET Z=0
 2240 LET K=A(Z)
 2250 IF K=10 THEN 2300
 2260 IF K=100 THEN 2320
 2270 POKE 8624H,00H: CURSOR 22,
9: PRINT K;"   ";
 2280 GOSUB 3000
 2290 IF E=1 THEN 2340
 2300 LET Z=Z+1
 2310 GOTO 2240
 2320 CURSOR 22,9: PRINT "オケマセン"
;
 2330 RETURN
 2340 LET A(Z)=10
 2350 RETURN
 2500 REM END
 2510 CURSOR 20,11: PRINT "GAME
OVER"
 2520 CURSOR 20,13
 2530 STOP
 3000 REM REVERSE
 3010 LET L=PEEK(7E94H)
 3020 IF L=204 THEN  LET L=49,M=
45: GOTO 3040
 3030 LET L=45,M=49
 3040 LET E=0,T=B+K
 3050 IF PEEK(T)<>9 THEN  RETURN

 3060 LET U=-11
 3070 GOSUB 3300
 3080 LET U=-10
 3090 GOSUB 3300
 3100 LET U=-9
 3110 GOSUB 3300
 3120 LET U=-1
 3130 GOSUB 3300
 3140 LET U=1
 3150 GOSUB 3300
 3160 LET U=9
 3170 GOSUB 3300
 3180 LET U=10
 3190 GOSUB 3300
 3200 LET U=11
 3210 GOSUB 3300
 3220 RETURN
 3300 REM SUB REVERSE
 3310 LET P=T+U
 3320 IF PEEK(P)<>M THEN  RETURN
 3330 LET Q=P+U
 3340 IF PEEK(Q)=0 THEN  RETURN

 3350 IF PEEK(Q)=9 THEN  RETURN

 3360 IF PEEK(Q)=L THEN 3390
 3370 LET Q=Q+U
 3380 GOTO 3340
 3390 LET E=1,Q=Q-B
 3400 FOR O=K TO Q-U STEP U
 3410 LET J=B+O
 3420 POKE J,L
 3430 GOSUB 1500
 3440 NEXT O
 3450 RETURN
8200 01 00 00 21 0B 83 3E 2D
8208 BE C2 10 82 04 C3 17 82
8210 3E 31 BE C2 17 82 0C 23
8218 3E 59 BD C2 06 82 78 32
8220 FF 82 79 32 FE 82 C9
```

18

ゲーム相手をマイコンに

実戦マージャン教室

TK-80BSL II

若 槻 匡 志

このプログラムは，麻雀をマイコン相手にするものです。マイコンと一対一のゲームですが，マイコン相手ですので，切り方や上り方など自分の技量発揮に最適です。

本プログラムは，TK80BSのRAMを全て実装し，LEVEL-II BASICを使用します。プログラム（BASIC及び機械語）は**第18―1表**に示します。

機械語の部分は主にテンパイ及び切りハイの判定ロジックとして使用し，BASICからCALLするサブルーチン型式となっています。

ゲームの方法

BASIC及び機械語をロード後，RUNすると先手人間（画面下側），後手マイコン（画面上側）の順に4枚ずつ配パイします。

配パイが終わるとカーソルが止って先手14枚，及び後手13枚のパイの分類を行ない，パイコード（**第18―1図**）順に整列して画面書示します（**画面写真**参照）。

ステパイA-NR?のKEY inメッセージより先手の切りハイの指示をします。パイの下側のアルフ

ァベットAからNまでの内，切ろうとするパイに対応したアルファベット1文字をKEY inします。

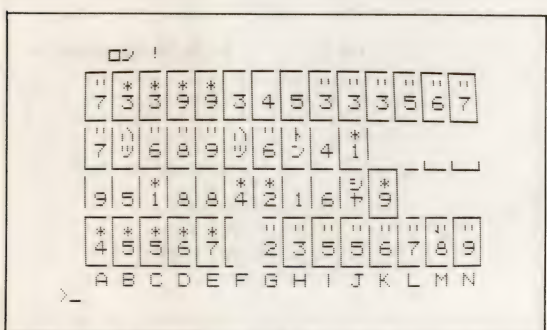
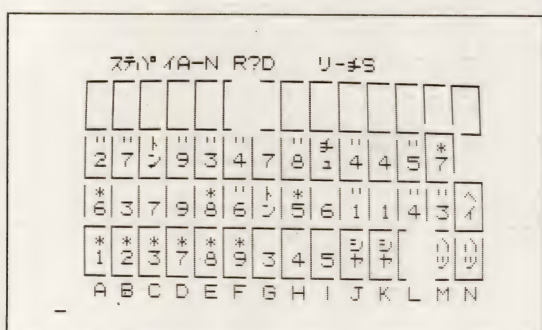
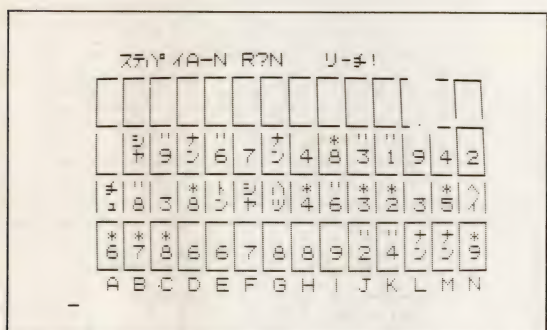
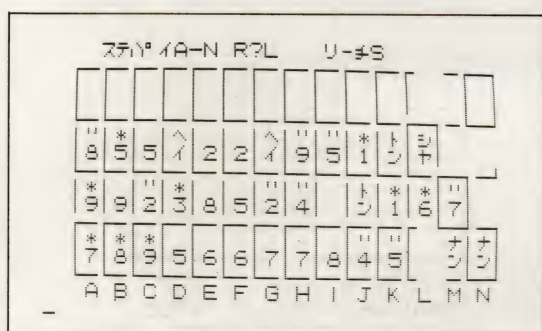
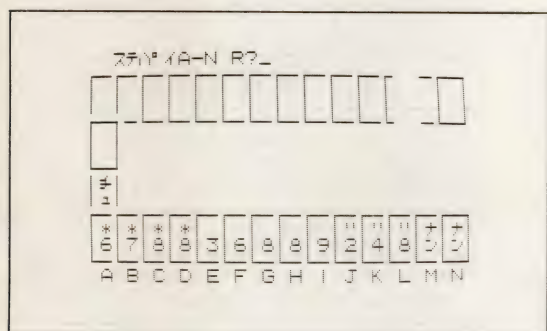
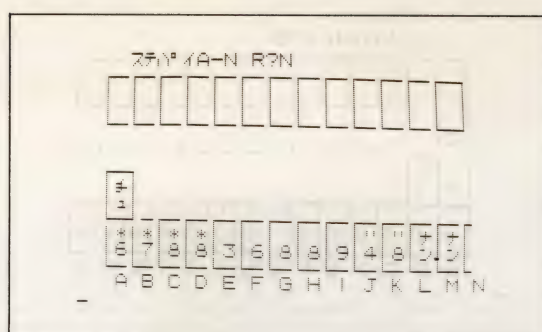
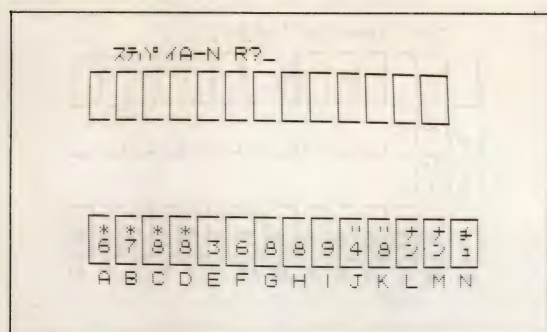
切ったパイはステパイとして，上の左側より順次表示されます。

次に，後手（マイコン）が上，右端14枚目の位置にパイをもって来て14枚分類後ステパイを判定し，前に1枚切ってきます。

先手，後手両者のうちいずれかがテンパイになるまで，ここまでの状態がくり返されます。

《第18－1図》パイの種類及びパイコード

先手がテンパイの状態になると，リーチ指示をします。

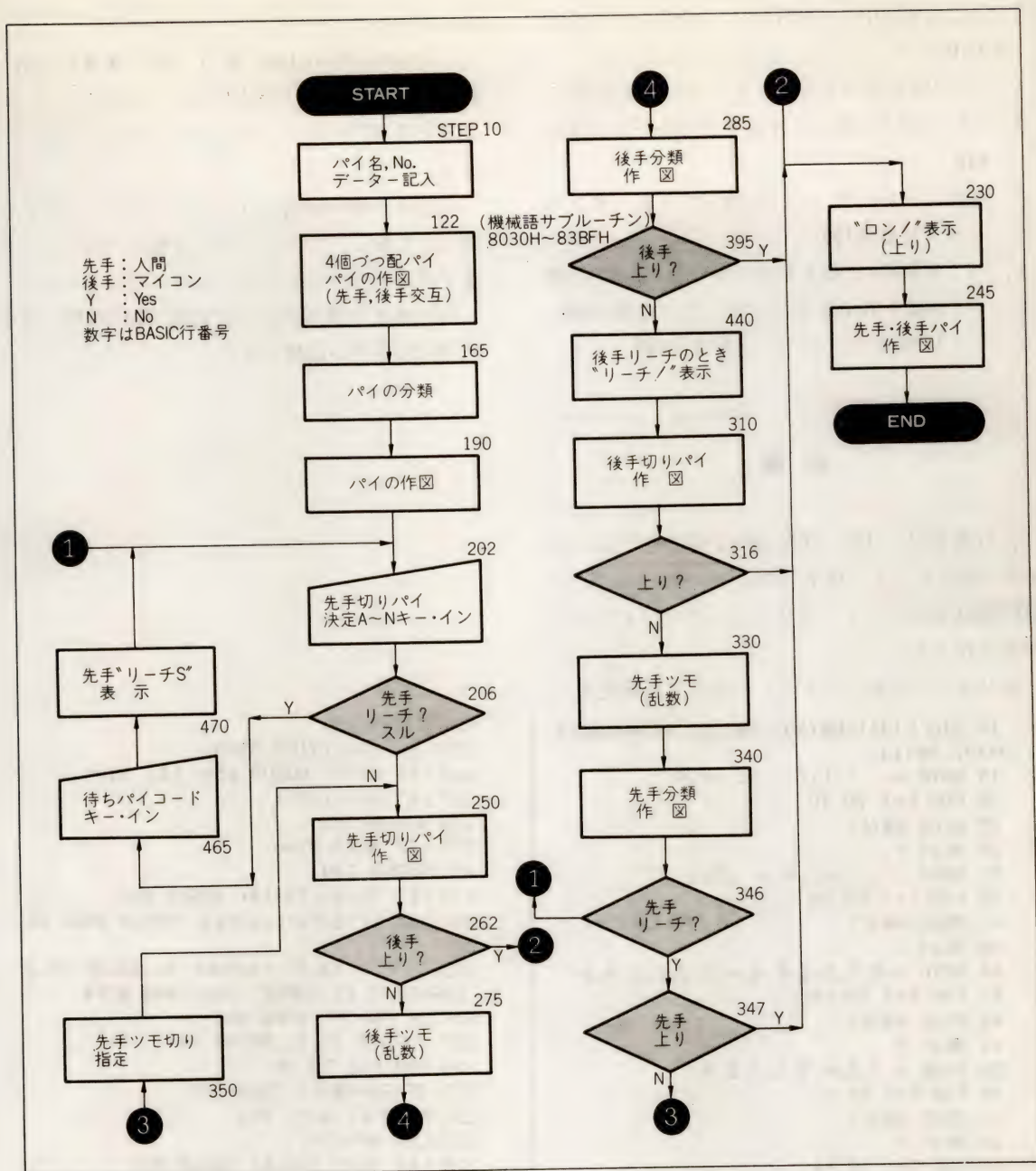
先手リーチの指示は，ステパイA-NR?のメッセージの時「R」をKEY inすると**マチパイKEY２?**のメッセージが表示されますので，待つべきパイ

コード(**第18—1図**)をKEY inします。(最大2個まで，1個待つ場合は2つ目は00を指定します)

先手のリーチ指示で画面に〝**リーチS**〟が表示されます。

先手のリーチ指示が行なわれると以後はロン(上がり)又は流れるまでオートマチックにゲームが進行します。

この間はコーヒーでも飲みながら観戦しましょう。

後手のリーチは機械語のロジックにより判定され，テンパイになると〝**リーチ!**〟の表示が現われます。

リーチ状態で先手，後手いずれかの切りパイ又はツモリパイが待ちパイに該当すると〝**ロン!**〟

《第18—2図》フローチャート

を表示し，両者のパイを全部表示後一回戦が終わります。

(後手のマイコンが先手のステパイで上った場合，14枚目（右端）のパイは除外してみます)

流れは，先手が28枚切ったところでカーソルがストップ状態になります。

プログラムの説明

フローチャートは**第18—2図**のようになります。機械語の部分は後手（マイコン）の上がり又はテンパイの判定ロジックをサブルーチン化しています。機械語プログラムの先頭アドレス8030Hですがパイ14個分のコードのワークエリア及びテンパイ，上がりの判定スイッチ，待ちコードのエ

リアとして8000H～802FHまでを使用しています。

BASICのサブルーチンは次のとおりです。

1．パイ1個を表示するサブルーチン行番号**500**

2．パイを4個ずつ配パイするサブルーチン行番号**650**

3．乱数によるツモ（パイを持ってくる）サブルーチン行番号**700**

4．パイを整理(分類するサブルーチン行番号**730**

5．パイN個を表示するサブルーチン行番号**800**

6．パイ1個消すサブルーチン行番号**850**

応 用

1．行番号135, 160, 200, 275で指定したS1の値を0に修正すると，後手**（マイコン側）**のパイがOPEN状態になって，切りパイ，テンパイの状況が見られます。

最初はこの状態でマイコンの打ち方を研究されるのもよいでしょう。

2．行番号320を**CUR. 5, 1 : IN "ﾅｷY orN" FS**に入れ代え「Y」をKEY inした場合マイコンの切りパイを「ナク」（チイ又はポンする）事ができます。

画面表示等多少物足りない点があるかもしれません。しかし，マイコンとの対戦ゲームとしてはまず満足して，できるのではないでしょうか？

自称麻雀中級者の私との対戦では4回戦って私の3勝1敗程度の成績です。

```
  10 DIM P(34),M$(10),N$(10),S(14),G(14
),O$(9),R$(14)
  15 DATA *, ,",ﾄ,ﾅ,ｼ,ﾍ, ,ﾊ,ﾁ
  20 FOR F=1 TO 10
  25 READ M$(F)
  27 NEXT F
  30 DATA , , ,ﾝ,ﾝ,ﾔ,ｲ, ,ﾂ,ｺ
  35 FOR F=1 TO 10
  40 READ N$(F)
  45 NEXT F
  46 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N
  47 FOR F=1 TO 14
  48 READ R$(F)
  49 NEXT F
  50 DATA 1,2,3,4,5,6,7,8,9
  55 FOR F=1 TO 9
  60 READ O$(F)
  65 NEXT F
 100 FOR F=1 TO 34
 105 LET P(F)=4
 110 NEXT F: RANDOMIZE : POKE 801AH,00H
: POKE 801CH,00H
 115 LET H$="    ": POKE 8011H,00H: POKE
801CH,00H
 120 CLEAR : LET S1=0,S9=0,G9=0
 122 RANDOMIZE
 125 FOR Z=1 TO 3
 127 LET S2=1,Z2=5
 130 LET Y1=11: GOSUB 650
 132 LET S2=0
 135 LET Y1=2,S1=1: GOSUB 650: LET S1=0

 140 NEXT Z
 145 LET Z2=3,Z=4,Y1=11,S2=1
 150 GOSUB 650
 155 LET Z2=2,Y1=2,S2=0
 160 LET S1=1: GOSUB 650: LET S1=0
 165 LET N=14,S2=1
 170 GOSUB 730
 175 LET N=13,S2=0
 180 GOSUB 730
 190 LET Y1=11,Z2=14: GOSUB 800
 200 LET Y1=2,S1=1,Z2=13: GOSUB 800: LE
T S1=0
 202 CURSOR 13,1: PICTURE 20,20,20,20,2
0: CURSOR 5,1: INPUT "ｽﾃﾊﾟｲA-N R"F$
 206 IF F$="R" THEN 455
 207 CURSOR 20,1: PRINT H$
 210 FOR F=1 TO 14
 215 IF F$=R$(F) THEN 250
 220 NEXT F: GOTO 202
 222 LET N=G(Z-1)
 225 LET Y1=2,Z2=14: GOSUB 800
 230 LET H$="ﾛﾝ !": CURSOR 5,1: PRINT H
$
 245 LET Y1=2,Z2=14: GOSUB 800: STOP
 250 LET N=S(F),S8=N,S(F)=S(14),Z=F
 252 LET X=2*F+1,Y=11: GOSUB 850
 255 LET S9=S9+1,X=2*S9+1,Y=8: GOSUB 50
0
 260 IF S9=14 THEN  LET S9=0
 262 LET X=PEEK(801AH): IF X=0 THEN 275

 264 LET Y=PEEK(8010H): LET F=PEEK(8011
H): IF S8=Y THEN 230
 266 IF S8=F THEN  GOTO 230
 275 GOSUB 700: GOSUB 700: LET S1=1
 280 LET X=29,Y=2,G(14)=N,L2=N: GOSUB 5
00
```

郵便はがき

お手数ですが
20円切手を貼
ってお出し下
さい。

141

（受取人）

東京都品川区東五反田
一—十一—十五

電波新聞社
出版部 行

○印をおつけください

小社をすでにご存知でしたか	知っていた 知らなかった
小社の新聞広告をご覧になったことがありますか ある（新聞名　　　　　　）　ない	
あなたにとって小社のイメージは いい　わるい　ふつう かたい　やわらかい　ふつう 専門的　一般的　どちらともいえぬ	
ご購読新聞名は	
定期購読中の雑誌名は	

```
285 LET N=14,S2=0: GOSUB 730
290 LET Y1=2,Z2=14: GOSUB 800
295 LET S1=0: GOTO 375
310 LET G9=G9+1,X=2*G9+1,Y=5: GOSUB 50
0
315 IF G9=14 THEN  LET G9=0
316 LET X=PEEK(8024H): IF X=0 THEN  GO
TO 320
317 LET Y=PEEK(8025H): IF G8=X THEN 23
0
318 IF G8=Y THEN 230
320 LET F$="N"
325 IF F$="Y" THEN 355
327 IF F$="Z" THEN 225
330 GOSUB 700: GOSUB 700
335 LET X=29,Y=11,S(14)=N,L2=N: GOSUB
500
340 LET N=14,S2=1: GOSUB 730
345 LET Y1=11,Z2=14: GOSUB 800
346 LET X=PEEK(8024H): IF X=0 THEN 202

347 LET Y=PEEK(8025H): IF L2=X THEN 23
0
349 IF L2=Y THEN 230
350 FOR F=1 TO 14: IF L2=S(F) THEN 250

352 NEXT F
355 LET N=G8
360 LET X=2*G9+1,Y=5: GOSUB 850
365 LET G9=G9-1: GOSUB 335
375 FOR F=1 TO 9
380 LET X=G(F): LET Y=8*4096+F
385 POKE Y,X
390 NEXT F
392 FOR F=1 TO 14
394 IF L2=G(F) THEN 400
396 NEXT F
400 LET X=G(10): POKE 800AH,X
405 LET X=G(11): POKE 800BH,X
410 LET X=G(12): POKE 800CH,X
415 LET X=G(13): POKE 800DH,X
420 LET X=G(14): POKE 800EH,X
421 LET I=L2: POKE 801DH,I: POKE 8000H
,F
425 CALL 8030H
430 LET X=PEEK(801CH): IF X>0 THEN 225

435 LET X=PEEK(801AH): IF X=0 THEN 445

440 LET H$="ﾘｰﾁ!": CURSOR 20,1: PRINT
H$
445 LET I=PEEK(8000H)
450 GOTO 480
455 CURSOR 13,1: INPUT "ﾏﾁﾊﾟｲKEY2"F
460 POKE 8024H,F
465 CURSOR 13,1: INPUT "ﾏﾁﾊﾟｲKEY 2"F
470 POKE 8025H,F: LET H$="ﾘｰﾁS": GOTO
202
480 LET N=G(I),G8=N,G(I)=G(14)
485 LET X=2*I+1,Y=2: GOSUB 850
490 GOTO 310
500 IF N>27 THEN 532
505 IF N>18 THEN 525
510 IF N>9 THEN 520
515 LET J=1: LET K=N: GOTO 530
520 LET J=2: LET K=N-9: GOTO 530
525 LET J=3: LET K=N-18: GOTO 530
530 LET N$(J)=O$(K): GOTO 535
532 LET J=N-24
535 CURSOR X,Y: PICTURE AD,9A,AE
540 LET Y=Y+1: CURSOR X,Y
545 PICTURE 8B,20,8B
550 IF S1=1 THEN 560
555 LET X=X+1: CURSOR X,Y: PRINT M$(J)
: LET X=X+1: CURSOR X,Y: PICTURE 8B: LET
 X=X-2
560 LET Y=Y+1: CURSOR X,Y: PICTURE 8B,
20,8B
565 IF S1=1 THEN 575
570 LET X=X+1: CURSOR X,Y: PRINT N$(J)
: LET X=X+1: CURSOR X,Y: PICTURE 8B: LET
 X=X-2
575 LET Y=Y+1: CURSOR X,Y: PICTURE AF,
9A,B0
577 CURSOR 1,16
580 RETURN
650 FOR Z1=2 TO Z2
652 LET Y=Y1
655 LET X=8*(Z-1)+2*Z1-1
657 LET J=Z1-1+4*(Z-1)
660 GOSUB 700
663 IF S2=1 THEN  GOTO 665
664 LET G(J)=N: GOTO 666
665 LET S(J)=N
666 GOSUB 500
667 NEXT Z1
670 RETURN
700 LET N=RND(35): LET N=INT(N)
705 IF N=0 THEN 700
710 IF P(N)=0 THEN 700
715 LET P(N)=P(N)-1: RETURN
730 LET Z2=N-1
735 FOR I=1 TO Z2
740 LET Z1=I+1
745 FOR J=Z1 TO N
750 IF S2=0 THEN 775
755 IF S(I)<S(J) THEN 765
760 LET Z=S(I),S(I)=S(J),S(J)=Z
765 NEXT J
770 GOTO 790
775 IF G(I)<G(J) THEN 765
780 LET Z=G(I),G(I)=G(J),G(J)=Z: GOTO
765
790 NEXT I: RETURN
800 LET Z2=Z2+1
805 FOR Z=2 TO Z2
810 IF Y1=2 THEN 820
815 LET N=S(Z-1): GOTO 825
820 LET N=G(Z-1)
825 LET X=2*Z-1: LET Y=Y1: GOSUB 500
830 IF Y1=2 THEN 840
835 LET F=Z-1: LET X=X+1: CURSOR X,15:
PRINT #1,R$(F)
840 NEXT Z: RETURN
850 LET X=X+1,Y1=Y
855 FOR F=1 TO 4
860 LET Y=Y1+F-1: CURSOR X,Y: PICTURE
20,20
865 NEXT F: RETURN
```

```
8030  3A 1A 80 C6 00 CA 52 80 3A 1D 80
47 3A 10 80 B8
 8040  CA 4A 80 3A 11 80 B8 C2 AA 81 3E
01 32 1C 80 C3
 8050  AA 81 3E 00 32 1B 80 32 22 80 3E
01 32 00 80 01
 8060  00 00 21 00 80 4E 09 7E C6 00 CA
CB 80 32 1E 80
 8070  FE 1C C3 D6 82 3E 00 CD E1 82 00
00 CA DB 80 3A
 8080  1E 80 C6 01 CD 19 83 00 00 C2 DB
80 3A 1F 80 32
 8090  20 80 3A 1E 80 C6 02 CD 19 83 C2
DB 80 3A 1F 80
 80A0  32 21 80 3A 22 80 C6 01 32 22 80
21 00 00 3A 00
 80B0  80 6F 01 00 80 09 1E 00 73 3A 20
80 21 00 00 6F
 80C0  09 73 3A 21 80 21 00 00 6F 09 73
3A 00 80 C6 01
 80D0  32 00 80 FE 0E C2 5F 80 C3 12 81
21 01 80 1E 00
 80E0  3A 1E 80 57 0E 01 7E BA CA F6 80
23 0C 79 FE 0F
 80F0  CA CB 80 C3 E6 80 3A 00 80 B9 CA
EB 80 1C 7B FE
 8100  01 CA 0B 81 79 32 21 80 C3 A3 80
79 32 20 80 C3
 8110  EB 80 3A 22 80 FE 04 CA AD 81 D6
03 C3 68 83 3E
 8120  01 C3 93 83 01 00 00 21 00 80 3A
00 80 4F 09 7E
 8130  C6 00 CA 74 81 32 1E 80 0C 21 00
80 09 5E BB CA
 8140  4E 81 79 FE 0E CA 74 81 3A 1E 80
C3 38 81 21 1F
 8150  80 71 CD 3A 83 3A 1B 80 C6 00 C2
74 81 3A 22 80
 8160  21 0F 80 85 6F 3A 1E 80 77 C3 74
81 F2 DB 80 D6
 8170  01 C3 D6 82 3A 00 80 C6 01 FE 0E
32 00 80 C3 78
 8180  83 3A 22 80 FE 02 C2 05 82 21 0E
80 0E 0E 7E C6
 8190  00 C2 99 81 2B 0D C3 53 83 79 32
00 80 3A 1B 80
 81A0  C6 00 C2 AA 81 3E 01 32 1A 80 C9
00 00 21 01 80
 81B0  0E 01 06 00 7E C6 00 C3 9E 83 23
0C C3 B4 81 04
 81C0  78 FE 02 C3 A7 83 32 20 80 C3, BA
81 32 21 80 5F
 81D0  3A 20 80 BB C2 DD 81 32 1C 80 C3
AA 81 79 32 00
 81E0  80 3A 20 80 32 10 80 C3 A5 81 3E
0E 32 00 80 3E
 81F0  00 32 1A 80 32 1B 80 32 1C 80 C3
DC 82 3E 00 32
 8200  30 80 C4 83 80 3A 22 80 FE 01 CA
12 82 3E 01 32
 8210  1B 80 3E 01 32 00 80 3E 00 32 22
80 21 00 80 01
 8220  00 00 3A 00 80 4F 09 7E C6 00 CA
C1 82 32 1E 80
 8230  FE 1C F2 C1 82 32 1F 80 D6 01 32
1E 80 C3 60 83
 8240  CA C1 82 3A 1F 80 32 1E 80 C6 01
CD 19 83 C2 93
 8250  82 CD 3A 83 3A 1B 80 C6 00 C2 C1
82 3E 00 CD E1
 8260  82 C2 6F 82 3A 1E 80 D6 01 32 10
80 C3 89 81 3E
 8270  01 CD E1 82 00 C2 83 82 3A 1E 80
C6 02 32 10 80
 8280  C3 89 81 3A 1E 80 D6 01 32 10 80
C6 03 32 11 80
 8290  C3 89 81 3E 00 00 CD E1 82 00 00
CA C1 82 3A 1E
 82A0  80 C6 02 CD 19 83 00 00 C2 C1 82
CD 3A 83 3A 1B
 82B0  80 C6 00 C2 C1 82 3A 1E 80 C6 01
32 10 80 C3 89
 82C0  81 3A 00 80 C6 01 32 00 80 FE 0E
C2 1C 82 3E 01
 82D0  32 1B 80 C3 89 81 F2 DB 80 32 1F
80 D6 01 C3 0A
 82E0  83 16 03 21 1E 80 5E 21 03 83 C6
00 C2 FD 82 23
 82F0  7E BB CA 03 83 15 C2 EF 82 15 C3
03 83 21 06 83
 8300  C3 EF 82 C9 08 11 1A 01 0A 13 32
1E 80 3E 00 CD
 8310  E1 82 3A 1F 80 C3 B7 83 00 5F 01
0E 00 21 00 80
 8320  09 7E BB CA 32 83 0D C2 1D 83 3E
01 00 C6 00 C3
 8330  34 83 00 00 21 1F 80 71 C9 00 3A
00 80 4F 06 80
 8340  3E 00 02 3A 1F 80 4F 3E 00 02 3A
22 80 C6 01 32
 8350  22 80 C9 C2 8E 81 3A 1F 80 C3 9A
81 00 00 00 00
 8360  3E 00 CD E1 82 C3 40 82 CA 1F 81
32 1B 80 C3 1F
 8370  81 00 FE 06 C3 B0 83 00 C2 24 81
3A 1B 80 C6 00
 8380  CA 81 81 3A 22 80 FE 06 F2 71 83
3E 00 32 22 80
 8390  C3 05 82 32 00 80 3E 00 32 22 80
C3 24 81 32 1E
 83A0  80 C2 BF 81 C3 BA 81 3A 1E 80 CA
CC 81 C3 C6 81
 83B0  CA AD 81 C3 D7 81 00 32 1E 80 CA
DB 80 C3 75 80
```

19

LEDだけで楽しめる

野球ゲームプログラム

H68/TR

匂坂哲次

このゲームはH68/TRのポケッタブル・コンソールの14桁表示機能を利用して行なう本格的野球ゲームです。

H68/TRの標準装備さえあれば，TVインターフェース等が無くとも手軽に遊べます。またプログラムは，1Kバイト以下にしてありますのでRAMの追加は一切不要です。

遊び方

このゲームは2人で遊びます。ゲームに先だち，先攻（Aチーム），後攻（Bチーム）をそれぞれ決めます。

プログラムがロードされたら，0 1 0 0 G の順にキーを操作するとゲームがスタートします（0100番地より始まります)。

すると，まず最初に，Aのように"SET"というメッセージを送ってきます。

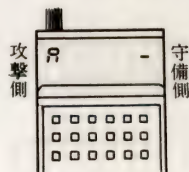
そうしたら先攻（Aチーム），後攻（Bチーム）の順に各々1回ずつ好きなキーを押してください（ただし，ABとRSのキーだけは押さないこと。又SF（シフトキー）を押した場合は引き続きもう一回押すことで1回分になります)。

これで，打撃の際の乱数発生用のワークエリアに入力されます。

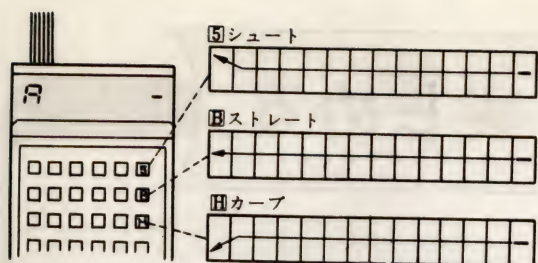
先攻，後攻と各1回ずつのキー入力が終わると，Bのように回数が表示されます。

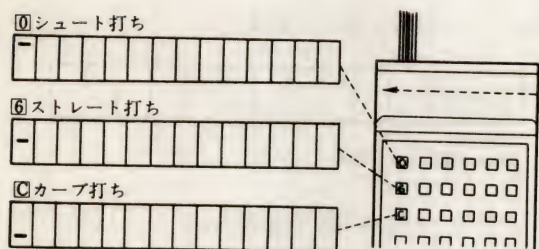
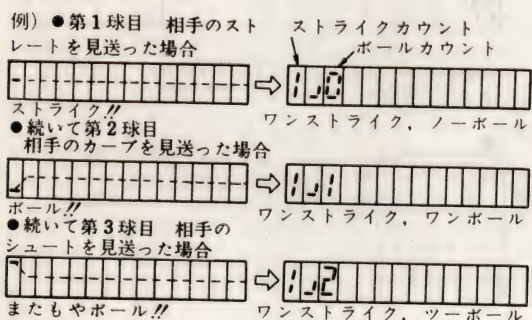
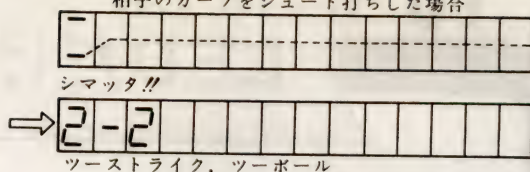
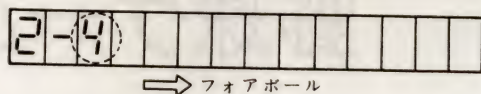
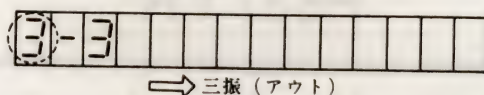
1回の表
(Aの攻撃)という意味

ここで，その回の攻撃側の人は，もう1回だけ好きなキーを押してください。前の2回とこの1回のキー入力で，乱数発生用のワークエリアの準備は完了です（この乱数の出方いかんにより，ヒットがでたり，アウトになったりが決まります)。

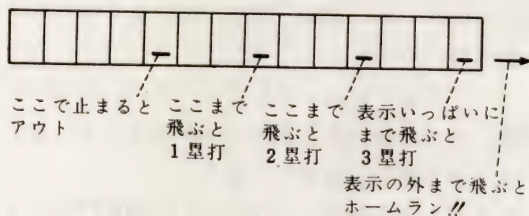
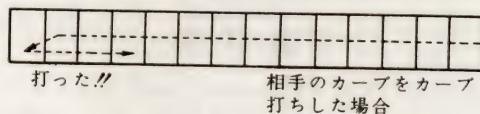
キー入力が終わると表示は次の様に変わります。

なお，2人のプレイヤーに対し，左図のような配置になるように，ポケッタブルコンソールを置いてください。

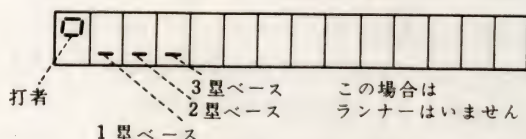
準備ができたら，守備側の人は，5，B，Hの三つのキーのどれかを攻撃側の人に見られないように，手でおおうか，何か壁状のものを立てるかして，隠しながらそっとキーを押してください。

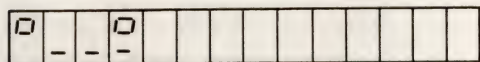
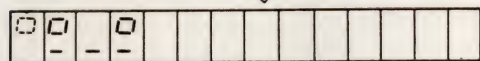
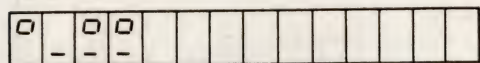
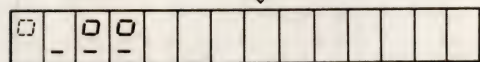
5，B，Hを押すことで，それぞれ，シュート，カーブ，ストレートが投げられます。

攻撃側の人は，その球を打とうと思ったら，球が自分のところへ届く（表示の一番左はしのところ）タイミングを合わせて，0，6，C，のキーのどれかを押してください。もちろん，打ちたくなければ見送ることもできますから，その場合はキーを押さないでください。

0，6，Cを押すことで，それぞれシュート打ち，ストレート打ち，カーブ打ちができます。

見送った場合には，相手がストレートを投げて来た場合のみストライクになります。カーブ及びシュートの場合はボールになります。球が止った状態を1秒間表示した後，その時点での投球カウントが表示されます。

相手の投げた球と，自分の打った位置とが違っていた場合は〝空振り〟となり，空振りの状態を表示した後，その時点での投球カウントが表示されます。〝空振り〟の場合はもちろんストライクカウントが一つ増えます。

ボールカウントが4になった場合はフォアボールとなります。

ストライクカウントが3になった場合は，三振（アウト）となります。

投球カウントが，まだ上の二つの例になっていない場合はまだ元の投球待ちの状態にもどります。

相手の投げた球と自分の打った位置がジャストミートした場合は，打球が飛んで行きます。

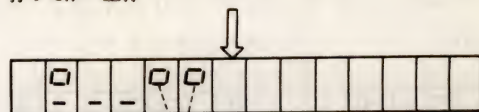
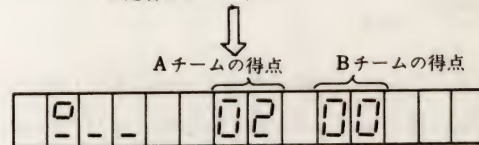
打球の飛んだ距離により，アウト，1塁打，2塁打，3塁打，ホームランのいずれかが決まります。

先述のフォアボールと三振，及び打った場合のアウト1塁打，2塁打，3塁打，ホームランのいずれかに決まると，表示が次の様に変わります。

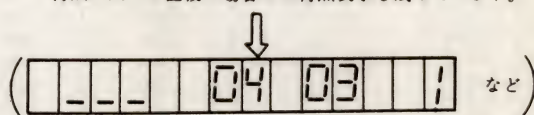
まず攻撃直前までの状態が表示されます（1秒間）。

続いて攻撃直後の状態が表示されます。

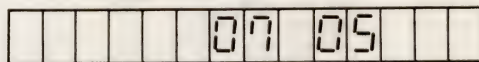
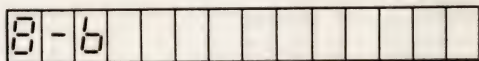
フォアボールの場合は打者が1塁へ進みます。もちろん，前の塁がふさがっていれば順にずれていきます。

三振及びアウトの場合には，打者のみ消えます。

ホームランの場合は打者及びランナーが四つ進みます。3塁打の場合は打者及びランナーが三つ進みます。

2塁打の場合は打者及びランナーが二つ進みます。

1塁打の場合だけ少し変わっていて，打つ前にもし2塁にランナーがいれば，このランナーは二つ進んでホームインすることができます。それ以外の場合は，打者ランナーとも一つだけ進みます。

ホームインした走者があった場合には，1秒後にその時点までの両チームの合計得点が表示されます。

引き続いて，その時点までのアウトカウント及びランナーの状況が表示されます。

得点の入った直後の場合には得点表示も残っています。

もし3アウトになっていなければ，この表示を2秒間した後，又元の〝投球待ち〟の状態にもどりますから，同じ要領でゲームを進めてください。

3アウトになった時は，2秒後に表示が消え，その回までの両チームの得点状況が表示されます。

その後，9回の裏までまだゲームが終わっていない場合には，引き続いて次の回数が表示されます。

そうしたら，又前と同様に攻撃側の人はキーを

1回押して乱数用の入力をしてください。

以下はまったく同じ方法で遊んでください。

ゲームが進んで9回の表まで終わるとその時点までの両チームの合計得点が表示された後〝END〟のメッセージが表示されます。

(12対9で
Aチームが勝ちました)

以上で1ゲームが終わりましたが，さらに続けて遊びたい場合には，何か好きなキーをひとつ押してください。

最初の〝SET〟の状態にもどりますので，以下全く同様に遊んでください。

応用

もう少し記憶容量を増せば〝盗塁〟〝バント〟〝ダブルプレー〟〝タッチアップ〟……etc. さらに本物の野球に近い複雑なゲームに展開させることが可能です。

しかし，現行のものでも素朴ではありますが，投・打のかけひきなどなかなか捨てがたい味があると思いますのでこのままでも十分に楽しめると思います。

アドレス	機械語	ラベル	オペレータ	オペランド	コメント
			ORG	$12	
0012		L01	RMB	1	回数カウンター
0013		L02	RMB	1	Aチームの得点カウンター（1位）
0014		L03	RMB	1	〃 （10位）
0015		L04	RMB	1	Bチームの得点カウンター（1位）
0016		L05	RMB	1	〃 （10位）
0017		L06	RMB	1	投球用ワークエリア
0018		L07	RMB	1	ストライクカウント用カウンター
0019		L08	RMB	1	ボールカウント用カウンター
001A		L09	RMB	1	アウトカウント用カウンター
001B		L0A	RMB	1	走者及び打者用のカウンター
001C		L0B	RMB	1	〃
001D		L0C	RMB	1	球の移動表示用ワークエリア
001E		L0D	RMB	1	
001F		L0E	RMB	1	乱数発生用ワークエリア
0020		L0F	RMB	1	
0021		L10	RMB	1	
0022		L11	RMB	1	得点チェック用ワークエリア
			ORG	$100	
0100	86 05		LDAA	#5	タイマー割込みを許す(表示のため)
0102	B7 E0 07		STAA	$E007	
0105	0E		CLI		
0106	BD F4 C3	L55	JSR	$F4C3	表示をクリア
0109	C6 93		LDAB	#$93	〝SET〟の表示
010B	F7 E8 14		STAB	$E814	
010E	C6 86		LDAB	#$86	
0110	F7 E8 15		STAB	$E815	
0113	C6 CE		JDAB	#$CE	
0115	F7 E8 16		STAB	$E816	
0118	BD F6 0D		JSR	$F60D	キー入力を記憶
011B	97 1F		STAA	L0E	
011D	BD F6 0D		JSR	$F60D	キー入力を記憶
0120	97 20		STAA	L0F	
0122	C6 02		LDAB	#2	回数カウンターインシャライズ
0124	D7 12		STAB	L01	
0126	7F 00 13		CLR	L02	Aチームの得点カウンタークリア
0129	7F 00 14		CLR	L03	
012C	7F 00 15		CLR	L04	Bチームの得点カウンタークリア
012F	7F 00 16		CLR	L05	
0132	C6 E8		LDAB	#$E8	球の移動の位置インシャライズ
0134	D7 1D		STAB	L0C	
0136	BD F4 C3	L53	JSR	$F4C3	表示をクリア
0139	96 12		LDAA	L01	

アドレス	機　械　語	ラ　ベ　ル	オペレータ	オペランド	コ　メ　ン　ト
013B	44		LSRA		
013C	BD 03 4C		JSR	L20	
013F	B7 E8 14		STAA	$E814	
0142	C6 BF		LDAB	#$BF	表示の回数
0144	F7 E8 15		STAB	$E815	
0147	BD 03 D3		JSR	L54	
014A	B7 E8 16		STAA	$E816	
014D	BD F6 0D		JSR	$F60D	キー入力を記憶
0150	97 21		STAA	L10	
0152	C6 01		LDAB	#1	
0154	D7 1B		STAB	L0A	走者及び打者用カウンター
0156	D7 1C		STAB	L0B	をイニシャライズ
0158	7F 00 1A		CLR	L09	アウトカウント用カウンターをクリア
015B	7F 00 18	L51	CLR	L07	ストライクカウント用カウンターをクリア
015E	7F 00 19		CLR	L08	ボールカウント用カウンターをクリア
0161	BD F4 C3	L35	JSR	$F4C3	表示をクリア
0164	BD 03 D3		JSR	L54	打者を表示
0167	B7 E8 14		STAA	$E814	
016A	C6 BF		LDAB	#$BF	投球用ボールを表示
016C	F7 E8 21		STAB	$E821	
016F	C6 21		LDAB	#$21	球の移動位置設定(最初の位置)
0171	D7 1E		STAB	L0D	
0173	BD F6 0D	L23	JSR	$F60D	投球キー入力を待つ
0176	81 35		CMPA	#$35	(シュートか？)
0178	27 0C		BEQ	L21	
017A	81 42		CMPA	#$42	(ストレートか？)
017C	27 0C		BEQ	L22	
017E	81 48		CMPA	#$48	(カーブか？)
0180	26 F1		BNE	L23	
0182	C6 F7		LDAB	#$F7	
0184	20 06		BRA	L24	
0186	C6 FE	L21	LDAB	#$FE	投げた球の種類を記憶
0188	20 02		BRA	L24	
018A	C6 BF	L22	LDAB	#$BF	
018C	D7 17	L24	STAB	L06	
018E	BD F4 C3	L27	JSR	$F4C3	
0191	DE 1D		LDX	L0C	
0193	C6 BF		LDAB	#$BF	
0195	E7 00		STAB	0X	
0197	BD 03 52		JSR	L25	打者の位置まで球を移動させる
019A	7A 00 1E		DEC	L0D	
019D	D6 1E		LDAB	L0D	
019F	C1 14		CMPB	#$14	
01A1	26 EB		BNE	L27	
01A3	BD F4 C3		JSR	$F4C3	
01A6	5F		CLRB		
01A7	F7 E0 06		STAB	$E006	
01AA	C1 06		CMPB	#6	
01AC	26 02		BNE	L28	打者はキーを押したかどうかチェッ
01AE	C6 FF		LDAB	#$FF	ク
01B0	5C	L28	INCB		
01B1	B6 E0 06		LDAA	$E006	
01B4	2B 26		BMI	L29	→見送り
01B6	81 00		CMPA	#0	
01B8	27 0A		BEQ	L2A	
01BA	81 10		CMPA	#$10	
01BC	27 0A		BEQ	L2B	
01BE	81 20		CMPA	#$20	
01C0	27 0A		BEQ	L2C	
01C2	20 18		BRA	L29	投球位置と打った位置を比較
01C4	C6 FE	L2A	LDAB	#$FE	
01C6	20 06		BRA	L2D	
01C8	C6 BF	L2B	LDAB	#$BF	
01CA	20 02		BRA	L2D	
01CC	C6 F7	L2C	LDAB	#$F7	
01CE	D1 17	L2D	CMPB	L06	
01D0	27 4C		BEQ	L2E	→打った!!
01D2	D4 17		ANDB	L06	

アドレス	機　械　語	ラ ベ ル	オペレータ	オペランド	コ メ ン ト
01D4	F7 E8 14		STAB	$E814	空振りの状態を1秒間表示する
01D7	BD F5 8C		JSR	$F58C	
01DA	20 13		BRA	L2F	→空振り（ストライク）へ
01DC	D6 17	L29	LDAB	L06	（見送りの場合）投球された球を1秒間表示する
01DE	F7 E8 14		STAB	$E814	
01E1	BD F5 8C		JSR	$F58C	
01E4	D6 17		LDAB	L06	投球はストレートだったか？
01E6	C1 BF		CMPB	#$BF	
01E8	27 05		BEQ	L2F	→ストライク
01EA	7C 00 19		INC	L08	ボールカウントをインクリメントする
01ED	20 03		BRA	L30	
01EF	7C 00 18	L2F	INC	L07	ストライクカウントをインクリメントする
01F2	96 18	L30	LDAA	L07	
01F4	BD 03 4C		JSR	L20	投球カウントを1秒間表示する。
01F7	B7 E8 14		STAA	$E814	
01FA	96 19		LDAA	L08	
01FC	BD 03 4C		JSR	L20	
01FF	B7 E8 16		STAA	$E816	
0202	C6 F3		LDAA	#$F3	
0204	F7 E8 15		STAB	$E815	
0207	BD F5 8C		JSR	$F58C	
020A	D6 18		LDAB	L07	ストライクカウントは3か？
020C	C1 03		CMPB	#3	
020E	26 02		BNE	L31	
0210	20 55		BRA	L32	→三振
0212	D6 19	L31	LDAB	L08	ボールカウントは4か？
0214	C1 04		CMPB	#4	
0216	26 03		BNE	L33	
0218	7E 02 8C		JMP	L34	→フォアボール
021B	7E 01 61	L33	JMP	L35	→投球待ち状態へ戻る
021E	BD 03 59	L2E	JSR	L36	アウトの位置まで打球を進める
0221	C1 19		CMPB	#$19	
0223	26 F9		BNE	L2E	
0225	BD 03 6B		JSR	L37	アウトの位置を突破したか？
0228	25 02		BCS	L38	
022A	20 3B		BRA	L32	→アウト
022C	BD 03 59	L38	JSR	L36	1塁打の位置まで打球を進める
022F	C1 1C		CMPB	#$1C	
0231	26 F9		BNE	L38	
0233	BD 03 6B		JSR	L37	1塁打の位置を突破したか？
0236	25 02		BCS	L39	
0238	20 35		BRA	L3A	→1塁打
023A	BD 03 59	L39	JSR	L36	2塁打の位置まで打球を進める
023D	C1 1F		CMPB	#$1F	
023F	26 F9		BNE	L39	
0241	BD 03 6B		JSR	L37	2塁打の位置を突破したか？
0244	25 02		BCS	L3B	
0246	20 17		BRA	L3C	→2塁打
0248	BD 03 59	L3B	JSR	L36	3塁打の位置まで打球を進める
024B	C1 22		CMPB	#$22	
024D	26 F9		BNE	L3B	
024F	BD 03 6B		JSR	L37	3塁打の位置を突破したか？
0252	25 02		BCS	L3D	
0254	20 06		BRA	L3E	→3塁打
0256	BD 03 59	L3D	JSR	L36	打球さらに進める(ホームラン)
0259	78 00 1B		ASL	L0A	(ホームラン)→打者及び走者をひとつ進める
025C	78 00 1B	L3E	ASL	L0A	(3 塁 打)→ 〃
025F	78 00 1B	L3C	ASL	L0A	(2 塁 打)→ 〃
0262	78 00 1B		ASL	L0A	〃
0265	20 41		BRA	L3F	→表示へ
					（アウト及び三振の場合）
0267	7A 00 1B	L32	DEC	L0A	打者のみクリアする
026A	7C 00 1A		INC	L09	アウトカウントをインクリメント
026D	20 39		BRA	L3F	→表示へ
					（1塁打の場合）
026F	78 00 1B	L3A	ASL	L0A	打者及び走者をひとつ進める
0272	D6 1B		LDAB	L0A	
0274	C4 08		ANDB	#8	
0276	27 12		BEQ	L40	
0278	D6 1B		LDAB	L0A	

アドレス	機械語	ラベル	オペレータ	オペランド	コメント
027A	C4 10		ANDB	#$10	
027C	26 06		BNE	L41	その時点で3塁に達している走者が
027E	D6 1B		LDAB	L0A	あれば，それをホームインさせる。
0280	CB 08		ADDB	#8	
0282	20 04		BRA	L42	
0284	D6 1B	L41	LDAB	L0A	
0286	CB 18		ADDB	#$18	
0288	D7 1B	L42	STAB	L0A	
028A	20 1C	L40	BRA	L3F	→表示へ
					（フォアボールの場合）
028C	D6 1B	L34	LDAB	L0A	
028E	C4 02		ANDB	#2	
0290	26 05		BNE	L43	
0292	7C 00 1B		INC	L0A	
0295	20 11		BRA	L3F	
0297	D6 1B	L43	LDAB	L0A	打者のみをひとつ進める，その場合
0299	C4 04		ANDB	#4	前の塁がつまっていたら，さらに前
029B	26 08		BNE	L44	の塁の走者も順に進める。
029D	D6 1B		LDAB	L0A	
029F	C6 03		ADDB	#3	
02A1	D7 1B		STAB	L0A	
02A3	20 03		BRA	L3F	
02A5	78 00 1B	L44	ASL	L0A	
02A8	BD 03 A5	L3F	JSR	L49	攻撃直前の打者と走者の状況を表示する。
02AB	D6 1B		LDAB	L0A	
02AD	D7 1C		STAB	L0B	打撃直後の打者と走者の
02AF	BD 03 A5		JSR	L49	状況を表示する
02B2	D6 1B		LDAB	L0A	
02B4	C4 0F		ANDB	#$0F	ホームインした走者の表示を消す
02B6	D7 1C		STAB	L0B	
02B8	BD 03 A5		JSR	L49	
02BB	D6 1B		LDAB	L0A	
02BD	C4 F0		ANDB	#$F0	ホームインした走者はあったか？
02BF	27 39		BEQ	L4A	
02C1	D6 1B		LDAB	L0A	
02C3	7F 00 22		CLR	L11	
02C6	58	L4D	ASLB		
02C7	24 25		BCC	L4B	
02C9	96 12		LDAA	L01	
02CB	44		LSRA		
02CC	25 11		BCS	L4C	
02CE	7C 00 13		INC	L02	
02D1	96 13		LDAA	L02	
02D3	81 0A		CMPA	#$A	
02D5	26 17		BNE	L4B	
02D7	7C 00 14		INC	L03	ホームインした走者の数をかぞえて
02DA	7F 00 13		CLR	L02	現在攻撃中のチームの得点カウンタ
02DD	20 0F		BRA	L4B	ーに加算する。
02DF	7C 00 15	L4C	INC	L04	
02E2	96 15		LDAA	L04	
02E4	81 0A		CMPA	#$A	
02E6	26 06		BNE	L4B	
02E8	7C 00 16		INC	L05	
02EB	7F 00 15		CLR	L04	
02EE	7C 00 22	L4B	INC	L11	
02F1	96 22		LDAA	L11	
02F3	81 04		CMPA	#4	
02F5	26 CF		BNE	L4D	
02F7	BD 03 B2		JSR	L4E	両チームの得点を表示する
02FA	96 1A	L4A	LDAA	L09	
02FC	BD 03 4C		JSR	L20	アウトカウントを表示する
02FF	B7 E8 21		STAA	$E821	
0302	D6 1A		LDAB	L09	
0304	C1 03		CMPB	#3	アウトカウントは3か？
0306	26 11		BNE	L4F	
0308	BD F5 8C		JSR	$F58C	1秒待つ
030B	BD F4 C3		JSR	$F4C3	表示をクリアする
030E	BD 03 B2		JSR	L4E	両チームの得点を表示する
0311	BD F5 8C		JSR	$F58C	2秒間待つ
0314	BD F5 8C		JSR	$F58C	
0317	20 12		BRA	L50	→回数のチェックへ

アドレス	機械語	ラベル	オペレータ	オペランド	コメント
0319	D6 1B	L4F	LDAB	L0A	走者及び打者用カウンターを次の打者の状態にセット
031B	C4 0F		ANDB	#$0F	
031D	5C		INCB		
031E	D7 1B		STAB	L0A	
0320	D7 1C		STAB	L0B	
0322	BD F5 8C		JSR	$F58C	2秒間待つ
0325	BD F5 8C		JSR	$F58C	
0328	7E 01 5B		JMP	L51	投球待ち状態へ戻る
032B	7C 00 12	L50	INC	L01	回数カウンターをインクリメント
032E	D6 12		LDAB	L01	10回の表になったか？
0330	C1 14		CMPB	#$14	
0332	27 03		BEQ	L52	→終わりへ
0334	7E 01 36		JMP	L53	→チェンジ
0337	C6 86	L52	LDAB	#$86	"END" を表示
0339	F7 E8 15		STAB	$E815	
033C	C6 AB		LDAB	#$AB	
033E	F7 E8 16		STAB	$E816	
0341	C6 A1		LDAB	#$A1	
0343	F7 E8 17		STAB	$E817	
0346	BD F6 0D		JSR	$F60D	キー入力待ち
0349	7E 01 06		JMP	L55	最初へ戻る
034C	8B 30	L20	ADDA	#$30	数字のセグメント化サブルーチン
034E	BD F5 11		JSR	$F511	
0351	39		RTS		
0352	CE 20 00	L25	LDX	#$2000	ディレー用サブルーチン
0355	09	L26	DEX		
0356	26 FD		BNE	L26	
0358	39		RTS		
0359	BD F4 C3	L36	JSR	$F4C3	球の移動表示サブルーチン
035C	DE 1D		LDX	L0C	
035E	D6 17		LDAB	L06	
0360	E7 00		STAB	0, X	
0362	BD 03 52		JSR	L25	
0365	7C 00 1E		INC	L0D	
0368	D6 1E		LDAB	L0D	
036A	39		RTS		
036B	D6 1F	L37	LDAB	L0E	乱数発生サブルーチン
036D	DB 20		ADDB	L0F	
036F	D7 20		STAB	L0F	
0371	DB 1F		ADDB	L0E	
0373	76 00 21		ROR	L10	
0376	D9 21		ADCB	L10	
0378	D7 1F		STAB	L0E	
037A	56		RORB		
037B	39		RTS		
037C	86 9C	L45	LDAA	#$9C	(ベースなし)打者及び走者の表示サブルーチン
037E	CE E8 14		LDX	#$E814	
0381	74 00 1C	L47	LSR	L0B	
0384	24 02		BCC	L46	
0386	A7 00		STAA	0, X	
0388	08	L46	INX		
0389	8C E8 1C		CPX	#$E81C	
038C	26 F3		BNE	L47	
038E	39		RTS		
038F	CE E8 15	L48	LDX	#$E815	ベースの表示追加サブルーチン
0392	E6 00		LDAB	0, X	
0394	C4 F7		ANDB	#$F7	
0396	E7 00		STAB	0, X	
0398	E6 01		LDAB	1, X	
039A	C4 F7		ANDB	#$F7	
039C	E7 01		STAB	1, X	
039E	E6 02		LDAB	2, X	
03A0	C4 F7		ANDB	#$F7	
03A2	E7 02		STAB	2, X	
03A4	39		RTS		
03A5	BD F5 8C	L49	JSR	$F58C	
03A8	BD F4 C3		JSR	$F4C3	
03AB	BD 03 7C		JSR	L45	
03AE	BD 03 8F		JSR	L48	

03B1	39		RTS		
03B2	96 13	L4E	LDAA	L02	両チームの得点表示サブルーチン
03B4	BD 03 4C		JSR	L20	
03B7	B7 E8 1B		STAA	$E81B	
03BA	96 14		LDAA	L03	
03BC	BD 03 4C		JSR	L20	
03BF	B7 E8 1A		STAA	$E81A	
03C2	96 15		LDAA	L04	
03C4	BD 03 4C		JSR	L20	
03C7	B7 E8 1E		STAA	$E81E	
03CA	96 16		LDAA	L05	
03CC	BD 03 4C		JSR	L20	
03CF	B7 E8 1D		STAA	$E81D	
03D2	39		RTS		
03D3	96 12	L54	LDAA	L01	攻撃チーム表示記号のセグメント化サブルーチン
03D5	84 01		ANDA	#1	
03D7	8B 41		ADDA	#$41	
03D9	BD F5 11		JSR	$F511	
03DC	39		RTS		
			END		

20 グラフ表示機能を！

バイオリズム・プログラム

TK-80BSL II

山県昌彦

人間の身体・感情・知性は，生まれた日を起点として，それぞれ23日，28日，33日を周期として，正弦曲線を描いて変化する，というのが，バイオリズムの理論です。

この理論にもとずいて得られる各指数やグラフをどう読みとるのかということは，バイオリズムに関する書物を見ていただくことにして，ここでは，生年月日と，調べたい年月日を入力すれば，その日の身体(P)，感情(F)，知性(I) の指数（+100～−100)と，その前後1か月のそれらの変化を示す曲線が画面に表示されるプログラムです。

プログラムの説明

生まれた日から調べたい日までの経過日数をDとします（生まれた日はD=0です)。

各指数は正弦曲線で表わされる変化をし，生まれた日にすべて0で出発するという理論ですから，たとえば，身体の指数は，周期が23日なので，

$$100\sin\left(\frac{2\pi D}{23}\right)$$

ですが，D=23×（整数)+D′ (0≦D′<23) とすれば，$100\sin\left(\frac{2\pi D'}{23}\right)$ でよいわけです。

さて，Dの計算ですが，ここでは，紀元1年1月1日を起点として，生まれた日までと，調べたい日までの日数の差として求めました。

うるう年については，4年1度を，1年の日数365.25日として，求める時点の前年末までの日数を結果の小数部を切捨てることによって求め，その年がうるう年で，求める日が3月以降ならば1を加えることによって正しく処理されます(800サブルーチン)。

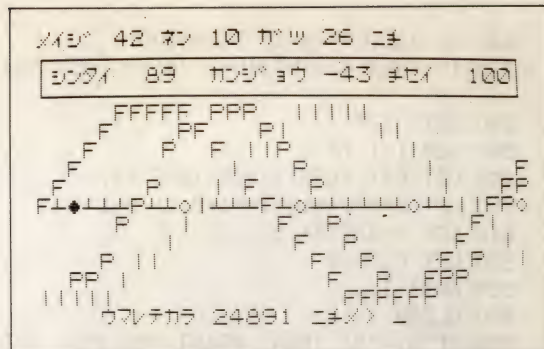
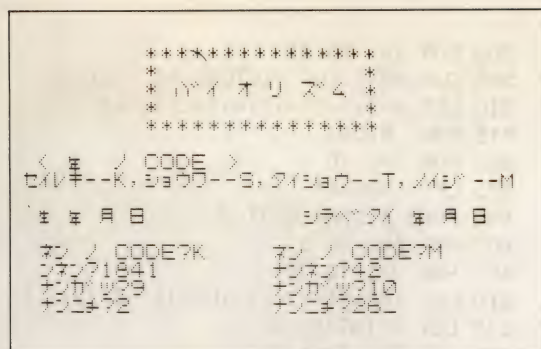
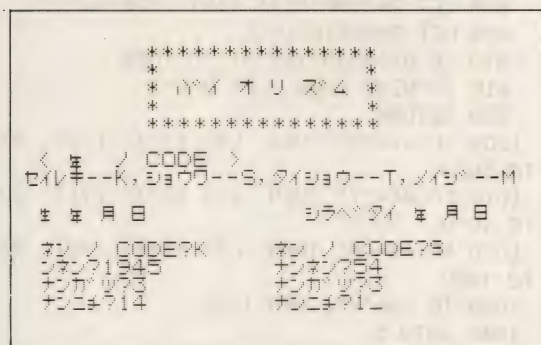
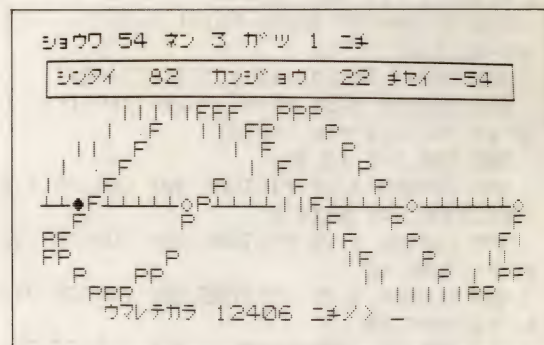
実は，年が4の倍数でも，100の倍数で400の倍数でない年（たとえば，1800年，1900年など）は，うるう年ではありません。この補正は，生まれた日と調べたい日との間に，そのような年の2月末が存在する場合のみ必要です。プログラム150～230がこの補正部分です。

たとえば，1799年に生まれた人の1900年3月以降の日のDは，1800年と1900年がうるう年でないため，前に求めたDより2を引いたものが，正しいDになります（200才以上の長寿は考えていません)。

A (B) は，1月1日からその年の (B—1) 月末までの平年の場合の日数です。

なお，TK-80BS LEVEL II BASICでは，有効数字の精度は6桁までですから，[365.25×(年数)]=365×(年数)+[0.25×(年数)]（[ ]内は小数部切捨て）とするところに注意してください。

```
           ***************
           *             *
           * バイオリズム *
           *             *
           ***************
< 年  ノ CODE  >
セイレキ--K,ショウワ--S,タイショウ--T,メイジ--M

生 年 月 日              シラベタイ 年 月 日

ネン ノ CODE?K          ネン ノ CODE?M
ナンネン?1841            ナンネン?42
ナンガツ?9              ナンガツ?10
ナンニチ?2              ナンニチ?26_
```

```
メイジ 42 ネン 10 ガツ 26 ニチ
シンタイ  89  カンジョウ -43 チセイ  100

ウマレテカラ 24891 ニチメ > _
```

この例は伊藤博文について，暗殺された日のバイオリズムを調べたものです（P：身体　F：感情　I：知性)。グラフの横軸は1目盛が1日，◆が調べる日として入力された日，◇は1週間ごとの印，P，F，Iの順に重ねて書くため，重なったところは前のが消されます。

データの入力

データの出力

バイオリズムプログラム・リスト

```
   5 CLEAR
   6 PRINT "          ***************"
   7 PRINT "          *             *"
   8 PRINT "          * バイオリズム *"
   9 PRINT "          *             *"
  10 PRINT "          ***************"
  15 PRINT " ": PRINT " < ";: PICTURE F
6
  20 CURSOR 6,7: PRINT " ノ CODE >"
  22 PRINT "セイレキ--K,ショウワ--S,タイショウ--T,メイ
ジ--M"
  24 CURSOR 2,10: PICTURE F5,20,F6,20,E
F,20,EE
  26 CURSOR 2,12: INPUT "ネン ノ CODE"A$
  28 INPUT "ナンネン"X1
  30 GOSUB 1000
  32 INPUT " ナンガツ"B
  34 INPUT " ナンニチ"C
  37 LET A=X1
  40 CURSOR 19,10: PRINT "シラベタイ ";: PI
CTURE F6,20,EF,20,EE
  41 CURSOR 17,12: INPUT "ネン ノ CODE"A$
  42 CURSOR 17,13: INPUT "ナンネン"X1
  43 GOSUB 1000
  45 CURSOR 17,14: INPUT "ナンガツ"B1
  47 CURSOR 17,15: INPUT "ナンニチ"C1
  50 LET A1=X1
  60 IF A1<A THEN 5
  65 IF A1<>A THEN 90
  70 IF B1<B THEN 5
  75 IF B1<>B THEN 90
  80 IF C1<C THEN 5
  90 DIM A(12),B(3),C(3),D(3),G(3)
  95 DATA 0,31,59,90,120,151,181,212,24
3,273,304,334
 100 FOR I=1 TO 12
 105 READ A(I): NEXT I
 110 LET B(1)=23,B(2)=28,B(3)=33,G(1)=1
6,G(2)=6,G(3)=9
 120 LET A2=A,B2=B,C2=C: GOSUB 800
 125 LET D1=D
 130 LET A2=A1,B2=B1,C2=C1: GOSUB 800
 135 LET D=D-D1
 150 IF A-400*INT(A/400)=0 THEN 190
 155 IF A-100*INT(A/100)=0 THEN 210
 160 LET N=INT(A/100)
 165 IF A1>100*(N+1) THEN  LET D=D-1: G
OTO 220
 170 IF A1<100*(N+1) THEN 300
 175 IF B1>2 THEN  LET D=D-1: GOTO 300
 180 GOTO 300
 190 IF A1<A+100 THEN 300
 195 IF A1=A+100 THEN 175
 200 LET D=D-1: GOTO 300
 210 IF B>2 THEN 190
 215 LET D=D-1: GOTO 190
```

```
 220 IF A1<100*(N+2) THEN 300
 225 IF (N+2-4)*INT((N+2)/4)=0 THEN 300

 230 GOTO 175
 300 FOR I=1 TO 3
 305 LET D(I)=D-B(I)*INT(D/B(I))
 310 LET X=100*SIN(2*PI*D(I)/B(I))
 315 LET X=INT(X+.5)
 320 LET C(I)=X
 325 NEXT I
 350 CLEAR
 355 IF A$="K" THEN  PRINT "ｾｲﾚｷ";: GOT
O 375
 360 IF A$="S" THEN  PRINT "ｼｮｳﾜ";: LET
 A1=A1-1925: GOTO 375
 365 IF A$="T" THEN  PRINT "ﾀｲｼｮｳ";: LE
T A1=A1-1911: GOTO 375
 370 IF A$="M" THEN  PRINT "ﾒｲｼﾞ";: LET
 A1=A1-1867
 375 PRINT A1;"ﾈﾝ";B1;"ｶﾞﾂ";C1;"ﾆﾁ"
 380 CURSOR 2,3: PRINT "ｼﾝﾀｲ ";C(1);" ｶ
ﾝｼﾞｮｳ ";C(2);"ﾁｾｲ ";C(3)
 385 FOR X=1 TO 31
 390 CURSOR X,2: PICTURE 9A: CURSOR X,4
: PICTURE 9A: NEXT X
 395 CURSOR 1,2: PICTURE AD: CURSOR 31,
2: PICTURE AE
 400 CURSOR 1,3: PICTURE 8B: CURSOR 31,
3: PICTURE 8B
 405 CURSOR 1,4: PICTURE AF: CURSOR 31,
4: PICTURE B0
 500 FOR X=1 TO 31
 505 CURSOR X,10: PICTURE A5: NEXT X
 510 LET A=((7*16+15)*16+2)*16+2
 515 POKE A,CAH
 520 FOR J=1 TO 4
 525 LET X=A+7*J
 530 POKE X,CCH: NEXT J
 600 FOR I=1 TO 3
 605 FOR U=1 TO 31
 610 LET X=5*SIN(2*PI*(U-D(I)-3)/B(I))
 615 LET X=INT(X+.5)
 620 LET Z=A-32*X+U-3
 625 POKE Z,G(I)
 630 NEXT U: NEXT I
 640 CURSOR 5,16: PRINT "ｳﾏﾚﾃｶﾗ";D+1;"ﾆ
ﾁﾒ";: END
 800 LET D=365*(A2-1)+INT(.25*(A2-1))
 805 LET D=D+A(B2)+C2
 810 IF A2-4*INT(A2/4)<>0 THEN 820
 815 IF B2>2 THEN  LET D=D+1
 820 RETURN
1000 IF A$="S" THEN  LET X1=X1+1925: GO
TO 1050
1010 IF A$="T" THEN  LET X1=X1+1911: GO
TO 1050
1020 IF A$="M" THEN  LET X1=X1+1867: GO
TO 1050
1030 IF A$="K" THEN 1050
1040 GOTO 5
1050 RETURN
```

21

画面処理の利用法

ミサイル・ゲーム

TK-80BSL I

高嶋康行

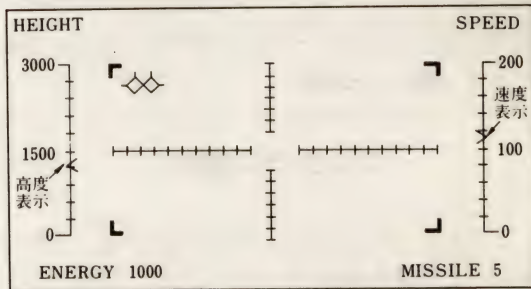
概要

このミサイル・ゲームは，TK-80BSで使用しているテレビディスプレイをミサイル・スコープとし，またキー・ボードを操縦桿として使用し，ミサイル・スコープ画面を見ながら，敵機の移動に合わせて高度，方向を操縦しながら，敵機をターゲット・スコープの中心に追い込みミサイルで打ち落とすゲームです。

このゲームはTK-80BSのLEVEL-1で記述されたプログラムで，使用されるメモリ容量はこのままでは5kワードをややオーバーしますので，マルチステートメントで書いてプログラムを短縮するか，TK-80BSのRAMをフル実装にして使用されるとよいでしょう。

ゲームの遊び方

このミサイル・ゲームは，テレビディスプレイ上に表示されるミサイル・スコープ（**第21－1図**）を見ながら，乗っているジェット機の機体を操作し，敵機をターゲット・スコープの中心に追い込み，ミサイルを発射し，撃墜した敵機の数を競うゲームです。

ミサイルの数は15本，エネルギーは7000です。敵機を深追いしすぎて，高度が下がると地上が現われて墜落します。上手に操縦して敵機を1機でも多く撃墜しましょう。

《第21－1図》ミサイル・スコープ画面

ゲームの終了は，次の様になっています。

1　地上に墜落した場合
2　ミサイルを使い果たした場合
3　エネルギーが無くなった場合

それぞれ，次のメッセージ

1　ツイラク　シマシタ
2　ミサイル　ヲ　ツカイハタシマシタ
3　ネンリョウ　ガ　ナクナリマシタ

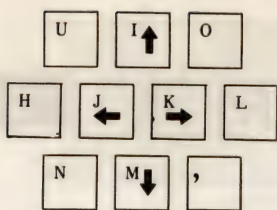
を画面上に表示し，撃墜した敵機の数を表示します。

第21－1図のミサイル・スコープ上で，

HEIGHT　：高度
SPEED　：速度
ENERGY　：残りエネルギー
MISSILE　：残りミサイル

を示します。

操縦は，TK-80BSのキー・ボードより行ないます。

キー・ボードは

《第21—3図》キー・ボードと方向の対応

I：上方へ（ディスプレイ上の敵機が下がる様に見える）

J：左へ（ディスプレイ上の敵機が右へ移動する様に見える）

K：右へ（ディスプレイ上の敵機が左へ移動する様に見える）

L：ミサイル発射

M：下方へ（ディスプレイ上の敵機が上昇する様に見える）

となっていて，キー・ボードを押して上手に操縦します（第21—2図)。

プログラム説明

ミサイル・ゲームは，大きく分けると六つのブロックより構成されています。

1　イニシャライズ
2　ミサイル・スコープの画面作成と表示
3　キー・ボード入力
4　位置の移動
5　ミサイル発射表示
6　ゲーム結果の表示

イニシャライズ　行番号　**1～50**

ミサイルゲームを行なうための初期値設定をします。

＊ミサイル数の設定：　B＝15
＊撃墜機数：　E＝0
＊エネルギー：　X＝7000
＊VIDEO RAMのアドレス：　S＝7E00H
＊ミサイル・スコープ表示用パターン：　配列エリア

ミサイル・スコープの画面作成と表示
行番号　**60～370**

ここでの処理は，まず高度計とスピード計の初期値設定をしています(高度　V＝1500，スピード　A＝120)。

ミサイル・スコープの画面作りは，PRINT-H文を使用しないで，POKE文により直接にVIDEO-RAMにパターン・データを書き込んで作成しています。

キー・ボード入力　行番号　**400～440**

TK-80BSは，I/Oのアドレスの割り付けとして，メモリー・マップドI/O方式を取っているため，7DFC番地をロードする事により，キー・ボードで押された文字に対応したコードが入力されます。

このプログラムでは，キー・ボードからの入力データの読み取りは，PEEK文で読み取り変数Zに格納しています。

このZの値により，乗っているジェット機の機体を上下左右に移動，ミサイルの発射かの判定を行ない，それぞれの処理ルーチンへジャンプさせます（第21—1表)。

位置の移動　行番号　**500～720，900～1106**

キー・ボードからの入力データにより，それぞれの処理を行なっています。

＊　上方向　（行番号500～520)
＊　左方向　（行番号600～620)
＊　右方向　（行番号700～720)
＊　下方向　（行番号900～915)

このプログラムでは，高度，位置，スピード等の表示は，G，H，I，J，K，L，V等の変数を用いて，POKE文により直接にVIDEO-RAMに書き込んでいます。

ミサイル発射表示　行番号　**800～890，2000，2020**

キー・ボードの〝L〞が押されるとミサイルが発射され，ミサイルの軌跡を表示します。ミサイルの軌跡もPOKE文にて処理を行なっています。

キー・ボード	コード(16進)	10　進	処　理
I	49H	73	上
J	4AH	74	左
K	4BH	75	右
L	4CH	76	ミサイル発射
M	4DH	77	下

《第21—1表》処理ルーチン

発射されたミサイルが敵機に命中すると（行番号860），その表示を行ない，撃墜機数を＋1します（行番号2000～2020）。

ゲーム結果の表示

ゲームの終了は，乗っているジェット機が墜落した場合，ミサイルを使い果たした場合，エネルギーがなくなった場合の3種類があり，それぞれにメッセージを表示し撃墜機数を表示します。

《第21—3図》ミサイル・フローチャート

HEIGHT　SPEED　000　1500　100　0　ENERGY 7000　MISSILE 15

◀エネルギー7000ミサイル15発でゲームスタート

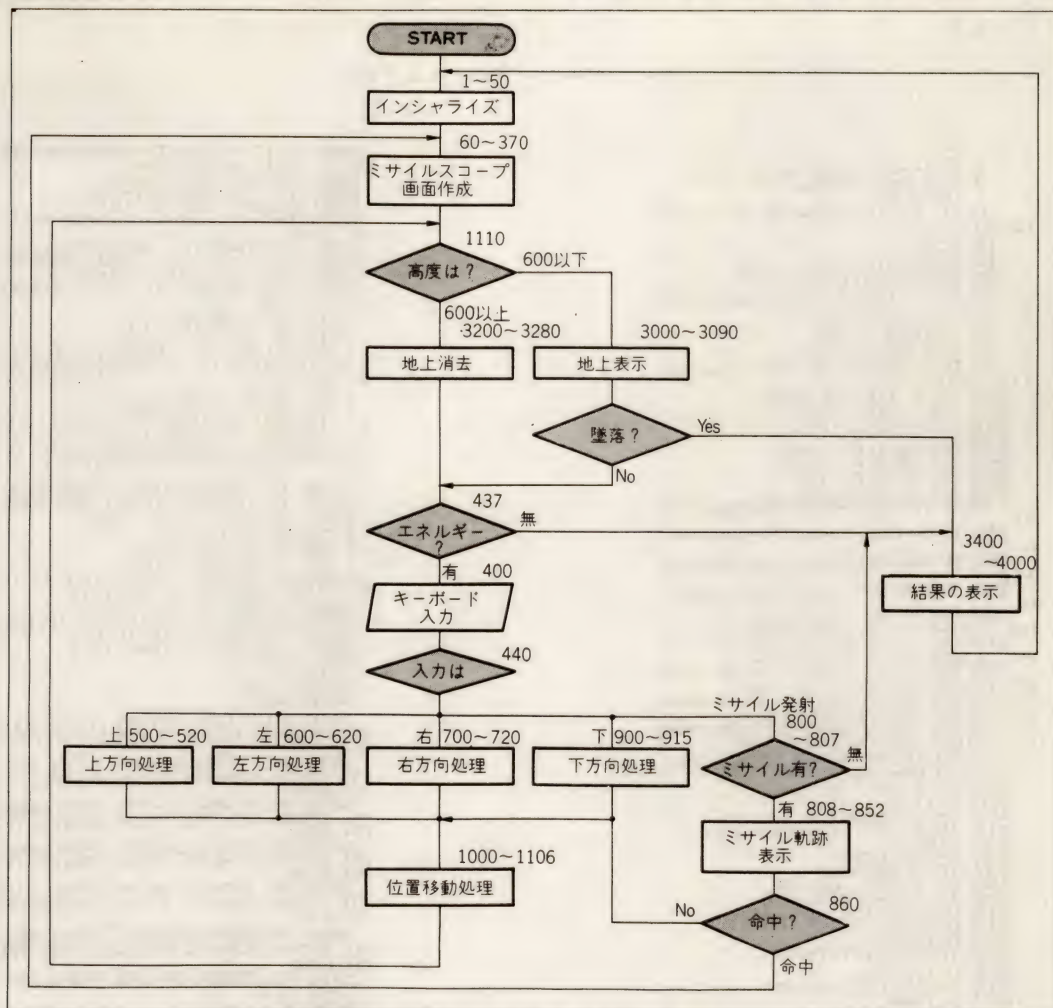
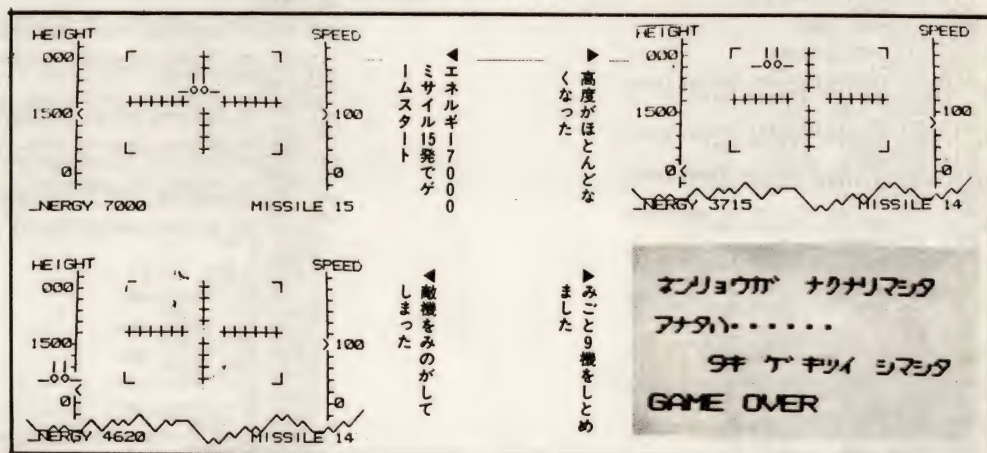
▶高度がほとんどなくなった

HEIGHT　SPEED　000　1500　100　0　ENERGY 3715　MISSILE 14

HEIGHT　SPEED　000　1500　100　0　ENERGY 4620　MISSILE 14

◀敵機をみのがしてしまった

▶みごと9機をしとめました

ネンリョウガ　ナクナリマシタ
アナタハ・・・・・
9キ　ゲキツイ　シマシタ
GAME OVER

フローチャート

第21－3図にミサイルゲームのフローチャートを示します。

プログラム・リスト

第21－3図にTK-80BSのLEVEL-1で記述したプログラム・リストを示します。

ミサイルゲーム・プログラムリスト

```
   1 CL.
   2 P."***** ﾐｻｲﾙ ｹﾞｰﾑ *****"
   3 FOR A=1 TO2000
   4 NEXT A
  10 CL. ;B=15;@(5)=192;@(7)=167
;@(0)=75
  11 C=0
  12 E=0
  15 @(20)=168,@(21)=190
  20 S=7*4096+14*256
  40 @(1)=139,@(2)=155,@(3)=204
  50 X=7000
  60 V=1500,A=120
  80 T=S+64+R.(176)
  90 G=T-32,H=T-31,I=T-1,J=T,K=
T+1,L=T+2
 200 CU. 1,1;P."HEIGHT"
 210 CU. 1,3;P."3000"
 220 CU. 1,8;P."1500"
 230 CU. 1,13;P."   0"
 240 CU. 28,1;P."SPEED"
 250 CU. 30,8;P."200"
 260 CU. 30,8;P."100"
 270 CU. 30,13;P."0"
 280 M=168,N=7*4096+14*256+2*16
+4
 285 F.N=NTON+384S.32;PO.N,M;N.
N
 290 M=168,N=7*4096+14*256+3*16
+12
 295 F.N=NTON+384S.32;PO.N,M;N.
N
 300 PO.7E49H,9EH;PO.7E57H,9FH;
PO.7F49H,A0H;PO.7F57H,A1H
 310 PO.7E50H,A4H;PO.7E70H,A4H;
PO.7E90H,A4H;PO.7EB0H,A4H
 320 PO.7EF0H,A4H;PO.7F10H,A4H;
PO.7F30H,A4H;PO.7F50H,A4H
 330 PO.7EC9H,A4H;PO.7ECAH,A4H;
PO.7ECBH,A4H;PO.7ECCH,A4H
 340 PO.7ECDH,A4H;PO.7ECEH,A4H;
PO.7ED2H,A4H;PO.7ED3H,A4H
 350 PO.7ED4H,A4H;PO.7ED5H,A4H;
PO.7ED6H,A4H;PO.7ED7H,A4H
 365 CU.1,16
 370 G.1110
 400 Z=P.(7DFCH)
 410 IF (Z<73)+(Z>77)Z=@(0)
 430 @(0)=Z
 435 X=X-R.(75)
 437 IF X<0G.3600
 440 G.(Z-72)*100+400
 500 IF T>S+479T=T-420
 510 U=1,T=T+32
 515 V=V+50
 520 G.1000
 600 IFT>S+510T=T-30
 610 U=1,T=T+1
 615 V=V+R.(25)-R.(25)
 620 G.1000
 700 IFT<S+34T=T+30
 710 U=1,T=T-1
 715 V=V+R.(25)-R.(25)
 720 G.1000
 800 IFU=0G.1000
 805 U=0
 807 IF B=0 G.3500
 808 B=B-1
 810 PO.7FE7H,D7H;PO.7FF8H,D7H;
PO.7FE7H,20H;PO.7FF8H,20H
 812 GOS. 890
 815 PO.7FC8H,D7H;PO.7FD7H,D7H;
PO.7FC8H,20H;PO.7FD7H,20H
 817 GOS.890
 820 PO.7FA9H,D7H;PO.7FB6H,D7H;
PO.7FA9H,20H;PO.7FB6H,20H
 822 GOS.890
 825 PO.7F8AH,D7H;PO.7F96H,D7H;
PO.7F8AH,20H;PO.7F96H,20H
 827 GOS.890
 830 PO.7FB6H,D7H;PO.7F75H,D7H;
PO.7FB6H,20H;PO.7F75H,20H
 832 GOS.890
 835 PO.7F4CH,D7H;PO.7F54H,D7H;
PO.7F4CH,20H;PO.7F54H,20H
 837 GOS.890
 840 PO.7F20H,D7H;PO.7F33H,D7H;
PO.7F20H,20H;PO.7F33H,20H
 842 GOS.890
 845 PO.7F0EH,D7H;PO.7F12H,D7H;
PO.7F0EH,20H;PO.7F12H,20H
 847 GOS.890
 850 PO.7EEFH,D7H;PO.7EF1H,D7H;
PO.7EEFH,20H;PO.7EF1H,20H
 852 GOS.890
 860 Z=P.(7ED0H);IF Z#32 G.2000
 870 G.1000
 890 R.
 900 IF T<S+64 T=T+410
 910 U=1,T=T-32
 915 V=V-50
1000 @(12)=G,@(13)=H,@(14)=I,@(
15)=J,@(16)=K,@(17)=L
1002 IF T<S+32 T=T+384
1003 IF T>S+479 T=T-384
1004 A=A+R.(10)-R.(10)
1005 T=T+(R.(2)-R.(2))*32+R.(2)
-R.(2)
1006 @(19)=@(18),@(6)=@(4)
1007 @(4)=S+420-(V+300)/300*32
1008 @(18)=S+444-A/20*32
1010 G=T-32,H=T-31,I=T-1,J=T,K=
T+1,L=T+2
1011 IF A<40 A=A+15
1012 IF A>185 A=A-15
1100 PO.@(12),M;PO.@(13),N;PO.@
(14),O;PO.@(15),P;PO.@(16),Q;PO.
@(17),R
1105 PO.@(6),@(7)
1106 PO.@(19),@(20)
1110 M=P.(G),N=P.(H);O=P.(I);P=
P.(J);Q=P.(K);R=P.(L)
1120 PO.G,@(1);PO.H,@(1);PO.I,@
(2);PO.J,@(3);PO.K,@(3);PO.L,@(2
)
1130 IF (V>600)*(C=1) GOS.3200
1150 IF (V<600)*(C=0) GOS.3000
1152 IF V>3100 V=V-100
1155 PO.@(4),@(5)
1156 PO.@(18),@(21)
1158 IF V<0 G.3400
1160 P."ENERGY",#4,X,;CU.22,16;
P."MISSILE",#2,B,;CU.1,16
1200 G.400
2000 PO.7EAFH,C9H;PO.7EB1H,C8H;
PO.7EEFH,C7H;PO.7EF1H,C6H
2005 FOR Z=1 TO1000 ;N.Z
2007 E=E+1
2010 C.
2020 G.60
3000 REM
3010 PO.7FC0H,B6H;PO.7FC1H,9DH;
PO.7FC2H,C1H;PO.7FC3H,C1H;PO.7FC
4H,9DH
3020 PO.7FC5H,B3H;PO.7FC6H,B6H;
PO.7FC7H,B3H;PO.7FC8H,BFH;PO.7FC
9H,BFH
3030 PO.7FAAH,C1H;PO.7FCBH,B6H;
PO.7FCCH,B3H;PO.7FCDH,BFH;PO.7FC
EH,96H
3040 PO.7FCFH,B6H;PO.7FF0H,B6H;
PO.7FF1H,C1H;PO.7FF2H,B3H;PO.7FF
3H,BFH
3050 PO.7FD4H,C1H;PO.7FD5H,C1H;
PO.7FD6H,B3H;PO.7FB7H,C1H;PO.7FB
8H,C1H
3060 PO.7FD9H,BFH;PO.7FDAH,B6H;
PO.7FDBH,9DH;PO.7FDCH,C1H;PO.7FD
DH,B3H
3070 PO.7FDEH,BFH;PO.7FBFH,B3H
3080 C=1
3090 R.
3200 PO.7FC0H,20H;PO.7FC1H,20H;
PO.7FC2H,20H;PO.7FC3H,20H;PO.7FC
4H,20H
3210 PO.7FC5H,20H;PO.7FC6H,20H;
PO.7FC7H,20H;PO.7FC8H,20H;PO.7FV
*3220 P8.7FAA1,20H;PO.7FCBH,20H;
PO.7FCCH,20H;PO.7FCD1,20H;PO.7FC
EH,20H
3230 PO.7FCFH,20H;PO.7FF0H,20H;
PO.7FF1H,20H;PO.7FF2H,20H;PO.7FF
3H,20H
3240 PO.7FD4H,20H;PO.7FD5H,20H;
PO.7FD6H,20H;PO.7FB7H,20H;PO.7FB
8H,20H
3250 PO.7FD9H,20H;PO.7FDAH,20H;
PO.7FDBH,20H;PO.7FDCH,20H;PO.7FD
DH,20H
3260 PO.7FDEH,20H;PO.7FBFH,20H
3270 C=0
3280 R.
3380 F.D=1 TO 5
3390 N.D;R.
3400 PO.7DFEH,00H
3405 GOS.3380
3410 PO.7DFEH,02H
3415 GOS.3380
3420 PO.7DFEH,00H
3425 GOS.3380
3430 PO.7DFEH,02H
3440 C.
3450 P."ﾂｲﾗｸｼﾏｼﾀ"
3460 G.4000
3500 C.;P."ﾐｻｲﾙ ｦ ﾂｶｲﾊﾀｼﾏｼﾀ"
3520 G.4000
3600 C.;P."ﾈﾝﾘｮｳｶﾞ ﾅｸﾅﾘﾏｼﾀ"
3620 G.4000
4000 P.;
4010 P."ｱﾅﾀﾊ・・・・・";P.;
4020 P.#3,E,"ｷ ｹﾞｷﾂｲ ｼﾏｼﾀ";P.;
4030 P."GAME OVER";P.;
4040 FOR A=1 TO 3000 ;NEXT A
4060 GOTO 1
```

22

マイコンの実用的活用

DATAファイルと検索

TRS-80

加藤敏秀

TRS-80のLEVEL II BASICを使ったデータファイルと検索のプログラムです。

まずプログラムを入力してRUNさせますと，最初に，

/NEW FILE

READY……CASSETTE FOR LOAD?<Y/N>

と表示されますから，すでにデータ・テープが準備されている場合はYを，またキー・ボードよりデータを入力する場合はNを押します。Nを押すと次に，

MENU<1/ENTER 2/LOAD 3/SAVE>?

が出ます。1はデータの入力を，2はデータのカセットからの入力を，3はデータをカセットに録音するためのものです

ここで2をENTERすればまた最初に戻ります。1をENTERしますとまだファイルの中味は空ですので，

/NEW FILE

/ENTER ITEM＃1?

と聞いてきます。ここでITEMとはデータの項目のことで，このプログラムは3個のITEMを持っています。ITEMには名前，住所，電話，コールサイン，品名，数量など自由に設定できます。ここでは月刊マイコンに掲載されているベーシックプログラムのデータ・ベースをつくってみます。

ITEM＃1には「雑誌名とページ」＃2には「プログラムの名前」＃3には「その他，使用機種など」を割り当ててみました。ITEM3個を入力し終わると，つづいて次からはDATA＃1から順番に各項目毎に聞いてきますからそれぞれ入力していきます。DATAの入力を打ち切る場合はITEM＃1の時にENTERキーだけを押せばOKです。例ではDATA＃15の入力のときにENTERキーだけ押してDATA入力を打ち切ったため，

/FILE CLOSED TO DATA＃14

と表示されて，14組のデータが格納されているのが確認されます。なおDATAの数はRAMのサイズによりますが，プログラムでは500組まで宣言されています。データの入力を打ち切ると，つづいて，

MENU<1/ENTER 2/LOAD 3/SAVE>?

の表示に戻ります。これは先程のMENUと同じ場所です。ここでENTERキーだけを押しますと検索SEARCHの作業に移れます。どんどんENTERキーだけを押しますと検索する項目は順次変わってきます。例ではITEM ＃2のNAMEの項目でPACHINKOと入力しました。ENTERすると検索が始まり，該当のDATA＃とITEM＃1の内容がすべて表示されます。

DATA＃5：MC-7808-23

これはDATA＃5がマイコン誌78年8月号23頁に該当のPACHINKOプログラムが見付かったことを表示しています。　つづいて

SEARCH DATA＃?

と聞いてきますから5をENTERすればDATA #5に関するすべての内容が表示されます。つづいて

CHANGE<1/BOOK 2/NAME 3/RMKS 4/CLEAR>?

と聞いてきます。1はITEM#1の内容を，2は#2, 3は#3の内容を変更したい場合，4はDATA #5の内容をすべて消去したい場合にENTERします。例では2をENTERしてプログラムの名前を変更しています。変更が終了すると新らしいDATA# 5の内容が表示されます。その他の検索の例としてITEM#3でTRSとENTERしてみます。この場合TRS-80まで入力することも80だけ入力することもOKです。ただし80だけではTK-80BSなども該当してくることは当然です。例ではTRSとENTERして検索した結果，

DATA #4：MC-7808-22

DATA #5：MC-7808-23

の2組が該当するデータとして見つかりました。これらのDATAの全体の内容は，

SEARCH DATA #?

の場合に4または5をENTERすればOKです。例では4をENTERしてDATA #4の内容を表示させています。

データをファイル

```
>RUN
/ NEW FILE
READY ... CASSETTE FOR LOAD ? < Y / N >
MENU < 1 / ENTER  2 / LOAD  3 / SAVE > ? 1
/ NEW FILE
/ ENTER ITEM # 1 ? BOOK
/ ENTER ITEM # 2 ? NAME
/ ENTER ITEM # 3 ? RMKS
/ DATA # 1
BOOK ? MC-7807-51
NAME ? WAKUSEI CHAKURIKU
RMKS ? TK-80BS L-I
/ DATA # 2
BOOK ? MC-7808-14
NAME ? MISSILE GAME
RMKS ? TK-80BS L-I
/ DATA # 3
BOOK ? MC-7808-18
NAME ? TEIDENATSU KAIRO NO SEKKE
RMKS ? HITACHI BASIC-I
/ DATA # 4
BOOK ? MC-7808-22
NAME ? KAZUATE GAME
RMKS ? TRS-80 L-I
/ DATA # 5
BOOK ? MC-7808-23
NAME ? PACHINKO GAME
RMKS ? TRS-80 L-I
/ DATA # 6
BOOK ? MC-7808-26
NAME ? YOUBI SAGASHI
RMKS ? NIBL
/ DATA # 7
BOOK ? MC-7808-27
NAME ? JANKYUU SIMULATION
RMKS ? NIBL
/ DATA # 8
BOOK ? MC-7808-28
NAME ? HANOI TOWER
RMKS ? TK-80BS L-I
/ DATA # 9
BOOK ? MC-7808-30
NAME ? SQUIGGLE
RMKS ? PET2001
/ DATA # 10
BOOK ? MC-7808-32
NAME ? ONGAKU JIDOU ENSOU
RMKS ? HITACHI MB6880
/ DATA # 11
BOOK ? MC-7808-36
NAME ? UCHUSEN TEKISENTOKI O KRIMETSU SEYO
RMKS ? LKIT-16
/ DATA # 12
BOOK ? MC-7809-14
NAME ? BOWLING GAME
RMKS ? TK-80BS L-I
/ DATA # 13
BOOK ? MC-7809-18
NAME ? KEIBA GAME
RMKS ? TK-80BS L-I
/ DATA # 14
BOOK ? MC-7809-86
NAME ? SIN COS
RMKS ? TK-80BS L-I
RMKS ? LKIT-16
/ DATA # 12
BOOK ? MC-7809-14
NAME ? BOWLING GAME
RMKS ? TK-80BS L-I
/ DATA # 13
BOOK ? MC-7809-18
NAME ? KEIBA GAME
RMKS ? TK-80BS L-I
/ DATA # 14
BOOK ? MC-7809-86
NAME ? SIN COS
RMKS ? TK-80BS L-I
/ FILE CLOSED TO DATA # 14
```

このプログラムは一種のロータリー式になっていて必要な作業の入口が見つかるまではENTERキーをどんどん押していけばOKです。

TRS-80のストリングは最大255文字まで使用できますのでデータの内容にかなり長い文献が入ります。文献の中の特定の単語を検索するのには大変便利でしょう。

このプログラムは欠点としてベーシックで作ったためスピードが遅いことが挙げられますが，TRS-80の豊富なEDIT機能を有効に利用されてさらに自分に適したプログラムに直される様期待します。しかし機能をあまり欲張ってプログラムを拡張しますと，RAMの残りが少なくなってDATAが制限されるのでご用心！ご用心！

データ検索

```
MENU < 1 / ENTER  2 / LOAD  3 /SAVE > ?
SEARCH DATA # ?
SEARCH BOOK ?
SEARCH NAME ?
SEARCH RMKS ?
MENU < 1 / ENTER  2 / LOAD  3 /SAVE > ?
SEARCH DATA # ?
SEARCH BOOK ?
SEARCH NAME ? PACHINKO
DATA # 5 : MC-7808-23
SEARCH DATA # ? 5
/ DATA # 5
BOOK : MC-7808-23
NAME : PACHINKO GAME
RMKS : TRS-80 L-I
CHANGE < 1 / BOOK   2 / NAME 3 / RMKS   4 / CLEAR > ?
CHANGE < 1 / BOOK   2 / NAME 3 / RMKS   4 / CLEAR > ? 2
NAME ? PACHINKO GAME ( AUTO )
/ DATA # 5
BOOK : MC-7808-23
NAME : PACHINKO GAME ( AUTO )
RMKS : TRS-80 L-I
MENU < 1 / ENTER  2 / LOAD  3 / SAVE > ?
SEARCH DATA # ?
SEARCH RMKS ? TRS
DATA # 4 : MC-7808-22
DATA # 5 : MC-7808-23
SEARCH DATA # ? 4
/ DATA # 4
BOOK : MC-7808-22
NAME : KAZUATE GAME
RMKS : TRS-80 L-I
CHANGE < 1 / BOOK   2 / NAME 3 / RMKS   4 / CLEAR > ?
SEARCH DATA # ?
SEARCH BOOK ?
SEARCH NAME ?
SEARCH RMKS ?
MENU < 1 / ENTER  2 / LOAD  3 /SAVE > ? 3
/ READY ... CASSETTE FOR SAVE ? < Y / N >
/ READY ... CASSETTE FOR SAVE ? < Y / N >
/ NOW SAVEING
/ CASSETTE SAVE OVER
MENU < 1 / ENTER  2 / LOAD  3 / SAVE > ? 2
READY ... CASSETTE FOR LOAD ? < Y / N >
/ NOW LOADING
```

```
10 'DATA BASE  BY J A 1 N U E
20 CLEAR8000:DEFSTRA:DEFINTB-Z:D
IMA(2,500):GOTO530
50 PRINT"/ DATA #"X:FORY=0TO2:PR
INTA(Y,0)" : "A(Y,X):NEXT:RETURN
80 PRINTCHR$(27)CHR$(27)CHR$(27)
90 A(0,N)="END OF FILE":PRINT"/
FILE CLOSED TO DATA #"N-1
100 PRINT"MENU  <  ENTER  /  LOA
D  /  SAVE  > ?"
102 A=INKEY$:IFA=""THEN102ELSEIF
A="E"THEN450ELSEIFA="L"THEN530EL
SEIFA="S"THEN660ELSE200
110 FORI=0TO2:PRINT"SEARCH "A(I,
0)" ";:A="":INPUTA:IFA=""NEXT:GO
TO100
120 Z=0:FORX=1TON-1:AA=A(I,X):FO
RY=1TOLEN(AA)-LEN(A)+1:IFA=MID$(
AA,Y,LEN(A))PRINT"DATA #"X": "A(
0,X):Z=1:Y=256
130 NEXTY,X:IFZ=0PRINT"/ NOT IN
FILE":GOTO110
200 X=0:INPUT"SEARCH DATA # ";X:
IFN=0PRINT"/ NEW FILE":GOTO100
210 IFX=0THEN110ELSEIFX>=NTHEN90
220 GOSUB50:PRINT"CHANGE  <";:FO
RI=0TO2:PRINTI+1"/ "A(I,0)"  ";:
NEXT:PRINT" 4 / DELETE >  ?"
222 A=INKEY$:IFA=""THEN222ELSEY=
VAL(A):ONYGOTO230,230,230,250:GO
TO200
```

```
230 Y=Y-1:PRINTA(Y,0)" ";:A="":I
NPUTA:IFA=""GOSUB50:GOTO100
240 A(Y,X)=A:GOSUB50:GOTO100
250 PRINT"/ DELETE DATA #"X": "A
(0,X)" < Y / N > ?"
252 A=INKEY$:IFA=""THEN252ELSEIF
A="Y"THEN254ELSE90
254 FORZ=XTON-1:FORI=0TO2:A(I,Z)
=A(I,Z+1):NEXTI,Z:N=N-1:GOTO90
450 IFN>0THEN470
460 PRINT"/ NEW FILE":FORI=0TO2:
PRINT"/ ENTER ITEM #"I+1;:INPUTA
(I,0):NEXT:N=1

470 PRINT"/ DATA #"N:FORI=0TO2:P
RINTA(I,0)" ";:A(I,N)="":INPUTA(
I,N):IFA(I,N)=""THFNAA

480 NEXT:N=N+1:IFN=500THEN80ELSE
470
530 IFN>0THEN550

540 N=0:PRINT"/ NEW FILE"

550 PRINT"READY ... CASSETTE FOR
 LOAD ? < Y / N >"
560 A=INKEY$:IFA=""THEN560ELSEIF
A="Y"THEN565ELSE100
565 PRINT"/ NOW LOADING"

570 INPUT#-1,A0,A1,A2,A3,A4,A5,A
6,A7,A8,A9,AA,AB,AC,AD,AE,AF,AG,
AH,AI,AJ,AK,AL,AM,AN,AO,AP,AQ,AR
,AS,AT,AU,AV,AW,AX,AY,AZ:A(0,N)=
A0:A(1,N)=A1:A(2,N)=A2:A(0,N+1)=
A3:A(1,N+1)=A4:A(2,N+1)=A5:A(0,N
+2)=A6:A(1,N+2)=A7:A(2,N+2)=A8:A
(0,N+3)=A9

580 A(1,N+3)=AA:A(2,N+3)=AB:A(0,
N+4)=AC:A(1,N+4)=AD:A(2,N+4)=AE:
A(0,N+5)=AF:A(1,N+5)=AG:A(2,N+5)
=AH:A(0,N+6)=AI:A(1,N+6)=AJ:A(2,
N+6)=AK:A(0,N+7)=AL:A(1,N+7)=AM:
A(2,N+7)=AN:A(0,N+8)=AO:A(1,N+8)
=AP:A(2,N+8)=AQ:A(0,N+9)=AR:A(1,
N+9)=AS

590 A(2,N+9)=AT:A(0,N+10)=AU:A(1
,N+10)=AV:A(2,N+10)=AW:A(0,N+11)
=AX:A(1,N+11)=AY:A(2,N+11)=AZ:FO
RX=NTON+11:IFA(0,X)="END OF FILE
"N=X:GOTO90

600 NEXT:N=N+12:IFN=500THEN80ELS
E570
660 PRINT"/ READY ... CASSETTE F
OR SAVE ? < Y / N >"
670 A=INKEY$:IFA=""THEN670ELSEIF
A="Y"THEN680ELSE100
680 PRINT"/ NOW SAVEING":X=0

690 A0=A(0,X):A1=A(1,X):A2=A(2,X
):A3=A(0,X+1):A4=A(1,X+1):A5=A(2
,X+1):A6=A(0,X+2):A7=A(1,X+2):A8
=A(2,X+2):A9=A(0,X+3):AA=A(1,X+3
):AB=A(2,X+3):AC=A(0,X+4):AD=A(1
,X+4):AE=A(2,X+4):AF=A(0,X+5):AG
=A(1,X+5):AH=A(2,X+5):AI=A(0,X+6
):AJ=A(1,X+6):AK=A(2,X+6)


700 AL=A(0,X+7):AM=A(1,X+7):AN=A
(2,X+7):AO=A(0,X+8):AP=A(1,X+8):
AQ=A(2,X+8):AR=A(0,X+9):AS=A(1,X
+9):AT=A(2,X+9):AU=A(0,X+10):AV=
A(1,X+10):AW=A(2,X+10):AX=A(0,X+
11):AY=A(1,X+11):AZ=A(2,X+11)
710 PRINT#-1,A0,A1,A2,A3,A4,A5,A
6,A7,A8,A9,AA,AB,AC,AD,AE,AF,AG,
AH,AI,AJ,AK,AL,AM,AN,AO,AP,AQ,AR
,AS,AT,AU,AV,AW,AX,AY,AZ:IFX>=NP
RINT"/ CASSETTE SAVE OVER":GOTO1
00
720 X=X+12:GOTO690

READY

>
```

23

工程管理への応用

PERT & TIME

TK80BSL II

近藤直之

このプログラムは，工程管理で，特に最近話題になっているPERT手法をプログラムしたものです。PERTについては，月刊マイコン'79 3月号"工程管理とPERT手法"の記事を参考にしてください。

プログラムの概略

イベントNo.，と所要日数を入力すればトータル・フロート等が計算されます。これによってクリティカル・パスを求めることができます。

また，目標とする最終日数を入力すればマイナスのトータル・フロートが表示され，日程短縮の目安となります。

画面の関係でTiL，FFは計算，表示しなかったのですが，あまり影響はないと思います。

また同じく画面の関係で，i，j，Tij，TFは2桁以内その他は3桁以内でないと表示が乱れます。

メモリ容量は9FFFをRAM ENDとした場合，配列にサンプル数50個分確保できます。マルチステートメントを用いることにより20個分くらい多くとれるようです。

プログラムの説明

行番号100〜300

始点のイベント (i)，終点 (j)，所要時間 (Tij) を入力します。このとき，メモリ容量 (配列) の関係でサンプル数は50個までと決められ，それ以上の入力は受け付けません (行250〜280)。また，I≧Jだと入力のやり直しを行ないます。(行180〜210)。入力終わりのときはIに1111と入力します (行150〜160)。

行番号300

終りの配列に終りの数字1111を代入します。

行番号400〜530

入力されたi，jを小さいものから順に並べ換えます。これは，日数計算のとき順に並んでいる必要があるからです。

行番号600〜950

順に並んだi，j，Tijをプリントします。この時サンプルが多くて画面一杯になり，自動的にスクロールされるのを防ぐため，10行ずつ表示し，OKか，訂正するのか，削除するのか，追加するのかを聞いて来ます。

行番号1200〜1310

削除のルーチンです。削除するNo.から後を一つずつ前へつめてリストのルーチンへ飛びます。

行番号1400〜2080

計算のルーチンです。

行番号1440〜1670

TiE (最早開始時刻) を計算します。

行番号1500：計算相手を捜します。

行番号1510：計算相手があったというフラグをセットします。

行番号1520：計算結果の最大値を捜します。

```
*** PERT/TIME ***
 サンプ゜ルハ 50コ マデ゛デ゛ス
 オワリ ノ トキハ I=1111 ト イレテ クダ゛サイ.
NO. 1
 I =?1
 J =?2
TIJ=?1

NO. 2
 I =?2
 J =?3
TIJ=?1_
```

```
 I    J  TIJ   TIE   TJE   TJL   TF
 1    2   1     0     1     1    0
 2    3   1     1     2     3    1
 2    4   2     1     3     3    0
 2    5   2     1     3     4    1
 3    5   1     2     3     4    1
-----------------------------------
 4    5   1     3     4     4    0
 5    6   1     4     5     5    0

_IST-AGAIN(1),AMEND(2),END(3),
SHORT(4)?3

 *** END ***
>_
```

PERT&TIMEデータ入出力

行番号1530：今までの最大値を入れ替えます。

行番号1550：相手がなければエラー表示へ飛びます。

行番号1560：この時のTiEに結果を代入します。

行番号1590〜1670

エラー表示，計算相手のない時エラーとなります。また計算上No.1は平行作業とはできません。

行番号1700〜1730

TjE（最早完了時刻）を計算します。

TjE＝Tij＋TiE

行番号1800〜2030：TjL（最遅完了時刻）を計算。

行番号1860：相手を捜します。

行番号1880：計算結果の最小値を捜します。

行番号1890：今までの最小値と入れ替えます。

行番号1910：相手がいなければエラー表示へ飛びます。

行番号1920：この時のTjLに結果を代入します。

行番号1950〜2030

エラー表示。計算相手のない時，エラーとなります。また計算上最後は平行作業とはできません。

行番号2050〜2080：TF（トータル・フロート）を計算します。TF＝TjL－TiE－Tij

行番号3000〜

計算結果のルーチンです。行番号600のリストと同様に10行ずつ表示してFUNCTIONの入力待ちとなります。

行番号3400

日程短縮のルーチンです。行番号3430で最後のTjLを強制的に目標とする最終日数に置き換え計算します。これにより日程短縮であればTFがマイ

プログラム・リスト

```
  10 CLEAR
  20 DIM A(50,6)
  30 PRINT "*** PERT/TIME ***":
PRINT
  40 PRINT " サンプ゜ルハ 50コ マデ゛デ゛ス"

 100 REM INPUT
 110 LET A=0
 120 PRINT
 130 PRINT " オワリ ノ トキハ I=1111 ト
イレテ クダ゛サイ."
 140 PRINT "NO. ";A+1
 150 INPUT " I ="A(A,0)
 160 IF A(A,0)=1111 THEN 300
 170 INPUT " J ="A(A,1)
 180 IF A(A,0)<A(A,1) THEN 220
 190 PRINT
 200 PRINT "  I<J デ゛ ナイト イケマセン"

 210 GOTO 140
 220 INPUT "TIJ="A(A,2)
 230 LET A=A+1: PRINT
 250 IF A<50 THEN 140
 260 PRINT "メモリー エンド゛"
 270 PRINT "  サンプ゜ルハ 50コ マデ゛デ゛ス
"
 280 PRINT "  リスト ヲ アゲ゛マス"
 300 FOR I=0 TO 6: LET A(A,I)=1
111: NEXT I
 400 REM SORTING
 410 PRINT
 420 PRINT " リストノ ナラベ゛カエ ヲ オコナッ
テイマス"
 430 PRINT "シハ゛ラク オマチクダ゛サイ"
 440 FOR I=0 TO A-2
 450 FOR J=I+1 TO A-1
 460 IF A(I,0)<A(J,0) THEN 520
 470 IF A(I,0)<>A(J,0) THEN 490

 480 IF A(I,1)<A(J,1) THEN 520
 490 LET B=A(I,0),C=A(I,1),D=A(
I,2)
 500 LET A(I,0)=A(J,0),A(I,1)=A
(J,1),A(I,2)=A(J,2)
 510 LET A(J,0)=B,A(J,1)=C,A(J,
2)=D
 520 NEXT J
 530 NEXT I
 600 REM LIST
 610 LET I=0
 620 CLEAR
 630 PRINT " NO.   I    J  TIJ"
 640 LET B=I+1: GOSUB 3520
 670 FOR J=0 TO 2
 680 LET B=A(I,J): GOSUB 3520
 690 NEXT J: PRINT
 760 IF A(I+1,0)=1111 THEN 880
 770 LET C=I-INT(I/10)*10
 780 IF C=9 THEN 810
 785 IF C=4 THEN  PRINT "------
----------"
 790 LET I=I+1
 800 GOTO 640
 810 PRINT
```

ナスになります。クリティカル・パスだけでなく，その他のアクティビティもマイナスになることが多く短縮の目安になります。

行番号3500～3540：表示の桁合わせルーチンです。

変数名

A：サンプル数

B：桁合わせ作業，ならべかえ作業

C：表示下位桁作業，ならべかえ作業，計算作業

D：ならべかえ作業

E：計算，MAX，MIN値

F：計算フラグ

I：一般作業

J：一般作業　　Z：FUNCTION

```
  820 INPUT "OK-NEXT(1),AMEND(2)
,ERASE(3),    PLUS(4)"Z
  830 IF Z=1 THEN  LET I=I+1: GO
TO 620
  840 IF Z=2 THEN 1000
  850 IF Z=3 THEN 1200
  860 IF Z=4 THEN 250
  870 GOTO 810
  880 PRINT
  890 INPUT "OK(1),AMEND(2),ERAS
E(3),PLUS(4),LIST-AGAIN(5)"Z
  900 IF Z=1 THEN 1400
  910 IF Z=2 THEN 1000
  920 IF Z=3 THEN 1200
  930 IF Z=4 THEN 250
  940 IF Z=5 THEN 600
  950 GOTO 880
 1000 REM AMEND
 1010 PRINT
 1020 INPUT "NO."Z
 1030 IF Z<1 THEN 1020
 1040 IF Z>A THEN 1020
 1050 INPUT " I ="A(Z-1,0)
 1060 INPUT " J ="A(Z-1,1)
 1070 IF A(Z-1,0)<A(Z-1,1) THEN
 1110
 1080 PRINT
 1090 PRINT " I<J デ ナイト イケマセン"
 1100 GOTO 1050
 1110 INPUT "TIJ="A(Z-1,2)
 1120 GOTO 400
 1200 REM ERASE
 1210 PRINT
 1220 INPUT "NO."Z
 1230 IF Z<1 THEN 1220
 1240 IF Z>A THEN 1220
 1250 FOR I=Z-1 TO A-1
 1260 LET A(I,0)=A(I+1,0)
 1270 LET A(I,1)=A(I+1,1)
 1280 LET A(I,2)=A(I+1,2)
 1290 NEXT I
 1300 LET A=A-1
 1310 GOTO 600
 1400 REM CALCULATE
 1410 PRINT
 1420 PRINT " タダイマ ケイサンヲ オコナッテ
イマス"
 1430 PRINT " シバラク オマチクダサイ"
 1440 REM TIE
 1450 LET C=0
 1460 LET A(0,3)=0
 1470 LET C=C+1,E=0,F=0
 1480 IF A(C,0)=1111 THEN 1700
 1490 FOR I=0 TO A-1
 1500 IF A(C,0)<>A(I,1) THEN 154
0
 1510 LET F=1
 1520 IF A(I,2)+A(I,3)<E THEN 15
40
 1530 LET E=A(I,2)+A(I,3)
 1540 NEXT I
 1550 IF F=0 THEN 1580
 1560 LET A(C,3)=E
 1570 GOTO 1470
 1580 PRINT
 1590 PRINT "ニュウリョク アヤマリ"
 1600 IF A(C,0)=A(0,0) THEN 1660
 1610 PRINT " I =";A(C,0);"ノ ケイサ
ンアイテガ アリマセン"
 1630 INPUT "OK-LIST(1)"Z
 1640 IF Z=1 THEN 600
 1650 GOTO 1630
 1660 PRINT " NO. 1 ハ ヘイコウサギョウト
シテハ イケマセン"
 1670 GOTO 1630
 1700 REM TJE
 1710 FOR I=0 TO A-1
 1720 LET A(I,4)=A(I,2)+A(I,3)
 1730 NEXT I
 1800 REM TJL
 1810 LET C=A-1
 1820 LET A(C,5)=A(C,2)+A(C,3)
 1830 LET C=C-1,E=1.E 11,F=0
 1840 IF C=-1 THEN 2050
 1850 FOR I=A-1 TO 0 STEP -1
 1860 IF A(C,1)<>A(I,0) THEN 190
0
 1870 LET F=1
 1880 IF A(I,5)-A(I,2)>E THEN 19
00
 1890 LET E=A(I,5)-A(I,2)
 1900 NEXT I
 1910 IF F=0 THEN 1940
 1920 LET A(C,5)=E
 1930 GOTO 1830
 1940 PRINT
 1950 PRINT "ニュウリョク アヤマリ"
 1960 IF A(C,1)=A(A-1,1) THEN 20
20
 1970 PRINT " J =";A(C,1);"ノ ケイサ
ンアイテガ アリマセン"
 1990 INPUT "OK-LIST(1)"Z
 2000 IF Z=1 THEN 600
 2010 GOTO 1990
 2020 PRINT " サイゴハ ヘイコウサギョウ ト
シテハ イケマセン"
 2030 GOTO 1990
 2050 REM TF
 2060 FOR I=0 TO A-1
 2070 LET A(I,6)=A(I,5)-A(I,3)-A
(I,2)
 2080 NEXT I
 3000 REM DISPLAY
 3010 LET I=0
 3020 CLEAR
 3030 PRINT "  I    J TIJ  TIE  T
JE  TJL  TF"
 3050 FOR J=0 TO 2
 3060 LET B=A(I,J): GOSUB 3520
 3070 NEXT J
 3100 FOR J=3 TO 5
 3110 LET B=A(I,J): GOSUB 3500
 3120 NEXT J
 3170 LET B=A(I,6): GOSUB 3520:
PRINT
 3190 IF A(I+1,0)=1111 THEN 3300
 3200 LET C=I-INT(I/10)*10
 3210 IF C=9 THEN 3240
 3215 IF C=4 THEN  PRINT "------
-------------------------"
 3220 LET I=I+1
 3230 GOTO 3050
 3240 PRINT
 3250 INPUT "NEXT(1),AMEND(2)"Z
 3260 IF Z=1 THEN  LET I=I+1: GO
TO 3020
 3270 IF Z=2 THEN 600
 3280 GOTO 3240
 3300 REM END
 3310 PRINT
 3320 INPUT "LIST-AGAIN(1),AMEND
(2),END(3),   SHORT(4)"Z
 3330 IF Z=1 THEN 3000
 3340 IF Z=2 THEN 600
 3350 IF Z=3 THEN 3370
 3355 IF Z=4 THEN 3400
 3360 GOTO 3310
 3370 PRINT
 3380 PRINT " *** END ***"
 3390 STOP
 3400 REM SHORT
 3410 LET C=A-1
 3420 PRINT
 3430 INPUT "END TIME"A(C,5)
 3440 PRINT
> _
```

24

実用プログラム

金種別仕訳の計算

TK-80BSL II

鈴木国宏

はじめに

私達が銀行などからお金を受取り，それをいくつかに分割する必要がある時，特に会社などにおいて給料を支払う時などは必ず会計係の人は金種別内訳書を添付して，給料袋に入れる時に，最も合理的に，かんじょう出来る金種で受領しています。この計算をマイコンでやってみようと考え，プログラムを組んでみました。

もっと単純になるかと思いますが，諸兄も考えてみて下さい。

使い方

プログラムをRUNさせると〝ナンニン〃？と，入力する人数を聞いてきますから，必要な人数をKEY-INします。このプログラムでは，90人がリミットです(私の勤務先の員数が85人なので)。もっと必要な方は行15のDIM　U (21)，Z (90) を変更して下さい。

帳簿上の支払一覧表氏名の頭にあらかじめ通し番号を打っておきます。人数をKEY-INすると，〝1〃〝キンガク〃？と聞いて来ますから，先程の一覧表ナンバーに従って，給料の金額をKEY-INします。

すべての入力が終わると〝データ　カクニンノタメ(1)ヲ　イレナサイ〃と表示されます。(1)以外をKEY-INしますと，すぐに計算に入ります。(1)をKEY-INすると入力したデータを1番から順に表示して来ます。もしデータがOKなら続けて(1)をKEY-INすれば次のデータを表示します。もしデータの誤りが発見されたなら，(1)以外をKEY-INして下さい。

〝テイセイチ　ヲ　イレナサイ〃
〝テイセイチ〃？

と聞いて来ますので，訂正値をKEY-INすれば，データを訂正して，次のデータから再度表示します。

データの確認がすべて終わると計算に入り

¥１０，０００　※　□□　＝　○○○エン
¥　５，０００　※　□□　＝　○○○エン
……………………………………………
……………………………………………

と表示され，下段に〝TOTAL (××エン)＋ (△△エン)〃と出ます。これは，レベル2　BASICでは有効数字が6桁までなので，数10万円の月給取りが7～8人で有効数字以下が切捨てられることになり，お金の計算上，極めて不都合ですので1000円以上の金種と，1000円未満とを，別個にTOTALしています。

この金額が，帳簿上の合計と一致しない時は，(1)以外をKEY-INして下さい。データ確認の段階にもどすことが出来ます。

```
ナンニンデ゛スカ?10
 1
キンカ゛ク?_
```

《写真24―1》データの入力

```
  3
キンカ゛ク?234512_
```

《写真24―2》金額の入力

プログラムの説明

10～60：データを読込む

行20のINPUT文で指定した人数分の金額をFOR……NEXT文のループで読込みます。

100～220：データの確認

行100のINPUT文で(1)以外をKEY-INすると，すぐに計算に入りますが，(1)をKEY-INすると，ふたたびFOR……NEXT文のループでデータを順次表示します。行160のINPUT文で(1)以外をKEY-INすると行200のINPUT文によりデータの訂正ができます。

300～500：金種の仕訳

FOR……NEXT文により，個人毎の金種仕訳と，それぞれの全員分を積算します。

550～580：金種別数量から金種別金額の換算

585～590：TOTAL金額の計算

1000円以上と未満に分けて計算します。

600～740：〝金種　※数量＝金額〟を表示します。

〝TOTAL（1000円以上のトータル）＋（1000円未満のトータル）〟を表示します。

```
1
          195675
         ********
テ゛ータ カ゛OK ナラ 1 ヲ イレナサイ?_
```

《写真24―3》データの確認

```
   10
キンカ゛ク?223456_
```

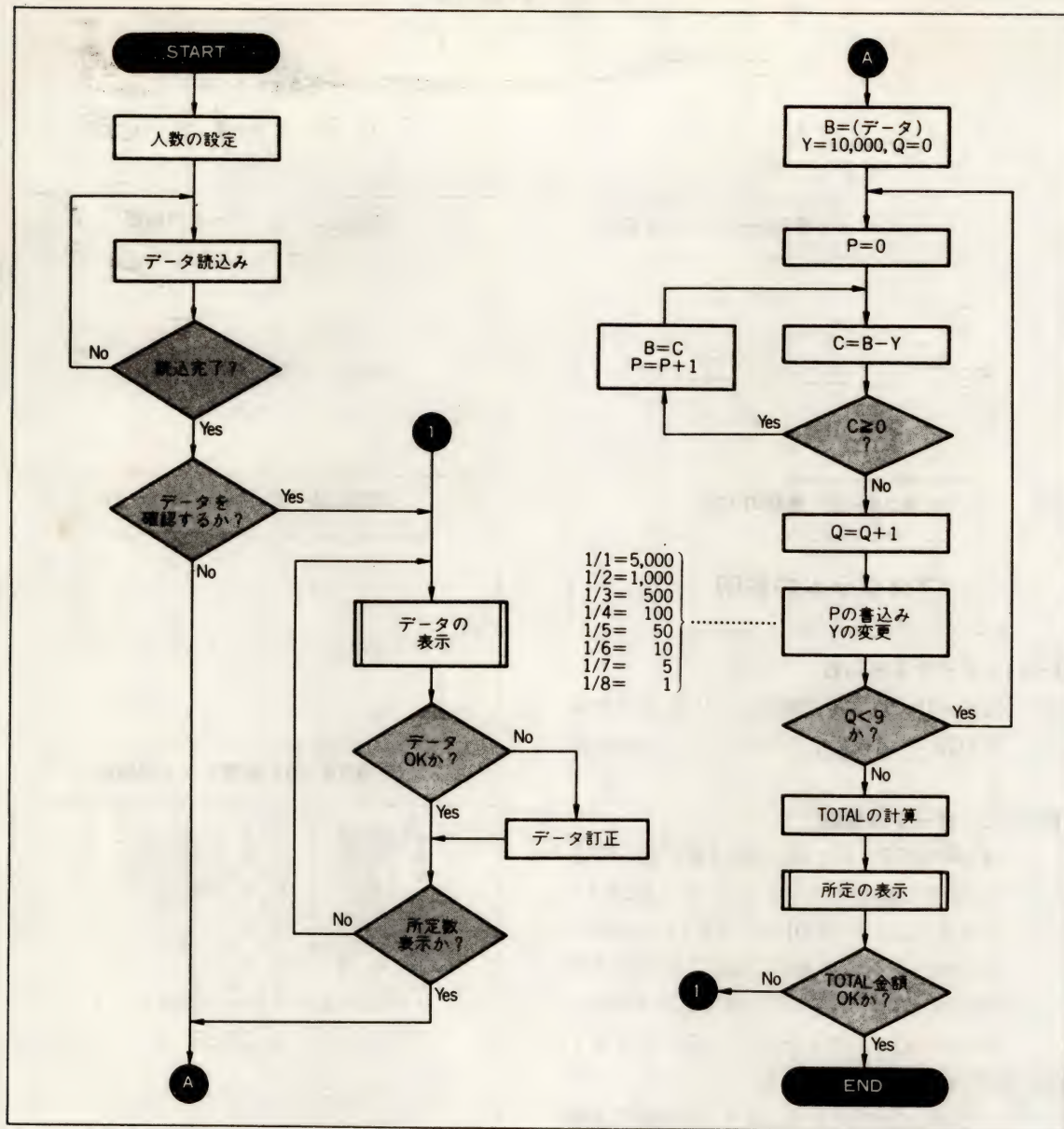
《写真24―4》金額を順に入力

```
テ゛ータ カクニン ノ タメ 1 ヲ イレナサイ?_
```

《写真24―5》金額データの確認

《写真24―6》結果の表示

750：INPUT文により帳簿とトータル値が異なる時は(1)以外をKEY-INして100にもどりデータの確認をする事が出来ます。

金種別仕訳計算フローチャート

```
 10 CLEAR
 15 DIM U(21),Z(90)
 20 INPUT "ﾅﾝﾆﾝﾃﾞｽｶ"H
 30 FOR N=1 TO H
 40 PRINT N: PRINT
 50 INPUT "ｷﾝｶﾞｸ"Z(N)
 60 CLEAR : NEXT N
 100 INPUT "ﾃﾞｰﾀ ｶｸﾆﾝ ﾉ ﾀﾒ 1 ｦ
ｲﾚﾅｻｲ"W
 110 IF W<>1 THEN  GOTO 221
 120 CLEAR
 130 FOR N=1 TO H
 140 PRINT N: PRINT TAB(8);Z(N)

 143 PRINT "          ********"
 145 PRINT
 150 INPUT "ﾃﾞｰﾀ ｶﾞOK ﾅﾗ 1 ｦ ｲﾚ
ﾅｻｲ"X
 170 IF X=1 THEN  GOTO 220
 175 PRINT "ﾃｲｾｲﾁ ｦ ｲﾚﾅｻｲ"
 180 PRINT
 181 INPUT "ﾃｲｾｲﾁ"Z(N)
 220 CLEAR : NEXT N
 221 CLEAR
 225 FOR N=11 TO 19
 230 LET U(N)=0
 240 NEXT N
 300 FOR N=1 TO H
 310 LET B=Z(N)
 320 LET Y=10000,P=0,Q=0
 330 LET P=0
 335 LET C=B-Y
 340 IF C<0 THEN  GOTO 400
 350 LET P=P+1
 360 LET B=C: GOTO 335
 400 LET Q=Q+1
 410 IF Q=1 THEN  LET U(1)=P: L
ET Y=5000: GOTO 330
 420 IF Q=2 THEN  LET U(2)=P: L
ET Y=1000: GOTO 330
 430 IF Q=3 THEN  LET U(3)=P: L
ET Y=500: GOTO 330
 440 IF Q=4 THEN  LET U(4)=P: L
ET Y=100: GOTO 330
 450 IF Q=5 THEN  LET U(5)=P: L
ET Y=50: GOTO 330
 460 IF Q=6 THEN  LET U(6)=P: L
ET Y=10: GOTO 330
 470 IF Q=7 THEN  LET U(7)=P: L
ET Y=5: GOTO 330
 480 IF Q=8 THEN  LET U(8)=P: L
ET Y=1: GOTO 330
 490 LET U(9)=P
 500 LET U(11)=U(11)+U(1): LET
U(12)=U(12)+U(2)
 510 LET U(13)=U(13)+U(3): LET
U(14)=U(14)+U(4)
 520 LET U(15)=U(15)+U(5): LET
U(16)=U(16)+U(6)
 530 LET U(17)=U(17)+U(7): LET
U(18)=U(18)+U(8)
 540 LET U(19)=U(19)+U(9): NEXT
 N
 541 LET J=U(16),K=U(17),L=U(18
),M=U(19)
 542 LET D=U(11),E=U(12),F=U(13
),G=U(14),I=U(15)
 550 LET U(11)=U(11)*10000: LET
 U(12)=U(12)*5000
 560 LET U(13)=U(13)*1000: LET
U(14)=U(14)*500
 570 LET U(15)=U(15)*100: LET U
(16)=U(16)*50
 580 LET U(17)=U(17)*10: LET U(
18)=U(18)*5
 585 LET U(20)=U(11)+U(12)+U(13
)
 590 LET U(21)=U(14)+U(15)+U(16
)+U(17)+U(18)+U(19)
 600 PRINT "***** ｷﾝｼｭ ｲﾁﾗﾝ ***
**"
 620 PRINT "¥10000 *",#1,D,"=",
#1,U(11)
 630 PRINT "¥ 5000 *",#1,E,"=",
#1,U(12)
 640 PRINT "¥ 1000 *",#1,F,"=",
#1,U(13)
 650 PRINT "¥  500 *",#1,G,"=",
#1,U(14)
 660 PRINT "¥  100 *",#1,I,"=",
#1,U(15)
 670 PRINT "¥   50 *",#1,J,"=",
U(16)
 680 PRINT "¥   10 *",#1,K,"=",
U(17)
 690 PRINT "¥    5 *",#1,L,"=",
U(18)
 700 PRINT "¥    1 *",#1,M,"=",
#1,U(19)
 720 PRINT "*******************
**"
 725 PRINT
 730 PRINT "TOTAL";TAB(8);U(20)
;TAB(16);"+";TAB(19);U(21)
 750 PRINT "ｺﾞｸﾛｳｻﾏ ﾃﾞｼﾀ!"
 760 END
> _
```

25

グラフィック機能アップ

TV画面作図プログラム

TK-80BS機械語

梶原好生

いつも思うのですがBSでゲームをやっているのをみると，うまく画面構成して楽しく演出してあります。

ですが，どうやってあんなパターンを描くのでしょうか。PETとは違ってBSでは厄介です。またBSにはPETと同等あるいは上回るパターンを持っていながら，それを十分に使いこなしているとは思えません。

大きな可能性がありながら誰もがうまく活用していないのではないでしょうか。そこで，画面に簡単にUSSエンタープライズなどを描けるプログラムを紹介します。

BSのゲームを演出するために

さて，私はときたま，名古屋の大須界隈に出かけてはこれらBASICマシンの前で，画面にディスプレイされる数々のゲームやグラフィックに見とれているのです。

しかし，その目はあまり好意に満ちたものではありません（この気持わかっていただけるでしょう）。

『あんなこと我達大なるTK-80BSでも簡単にできるわい！』

その意味ごみとはうらはらに，BSのパターンはプログラム中でしか使えないために，実際に画面で絵などを構成するためには，事前に紙の上でレイアウトしなければなりません。ところが，よく考えたつもりでも実際にできたパターンは期待はずれだったりずれてたり，くじけてしまいます。

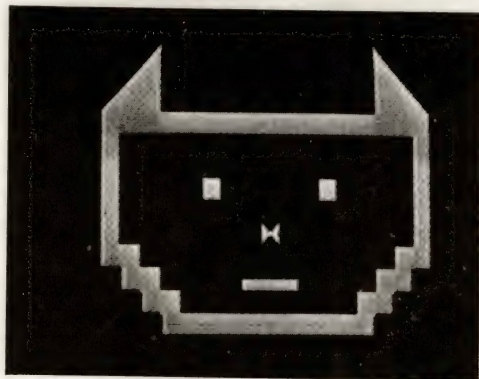
でも楽しいゲームを演出するためにはこれは必要不可欠です。それではあの悪名高い（と私が勝手に対抗上決めた）PETではどうでしょうか？

名古屋コスモスの奥村電機さんでPETにさわらせてもらえたのでさっそくスパイしてみると，なんとキーから直接パターンを指定できる上にカーソルを任意の場所に動かせるではありませんか（カーソル移動はただ単にグラフィックのためだけじゃないんだけど……)。

そこで再び『こんなことBSでも簡単にできるわい！』

そんなわけで次のプログラムを作りました。

グラフィック・プログラム

PETとBSとは本格的に違うものですから，ど

《写真25-1》こんな画面が自由に

←	→	↑	↓	RESET
C	D	E	F	全画面消去
8	9	A	B	行を消去
4	5	6	7	画面退避
φ	1	2	3	画面復帰

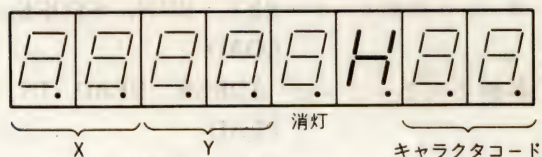
《第25-1図》TK-80キーの操作

うしてもPETほど使い勝手はよくなりませんか，パターンに関しては，BSはPETのパターンを含み，それ以上のものを持っているので十分対抗できると思います。

マイコン界の一つの現象として，BASICで入門して次にアセンブラに進むという人が今年からは増えていくことと思います。古い方には面白くないことかも知れませんが，特にリアルタイムの処理にBASICから機械語へリンクしなければならないという必然から，自然にアセンブラに進んでいくのが一つの理想だと思います。そのときこのモニタはしっかりとこういった人たちを受けとめてくれるでしょう。

グラフィックを簡単に作るために

まず，キー一つでカーソルの位置がわかること，それから描いたパターンのコードがわかることがグラフィックを簡単につくるために必要です。

最初の要素をみたすためには，まず問題になる

CURSOR X.Y; PRINT H○○

《第25-2図》LEDの表示

のは，どのキーを用いるかです。これは母なるTK-80の16進キー・ボードを用いましょう。次の二つめの要素これはどこに表示するのでしょうか？そうですTK-80のLEDを用いればよいのです。

というわけで，すべて母なるTK-80の厄介になることにします。

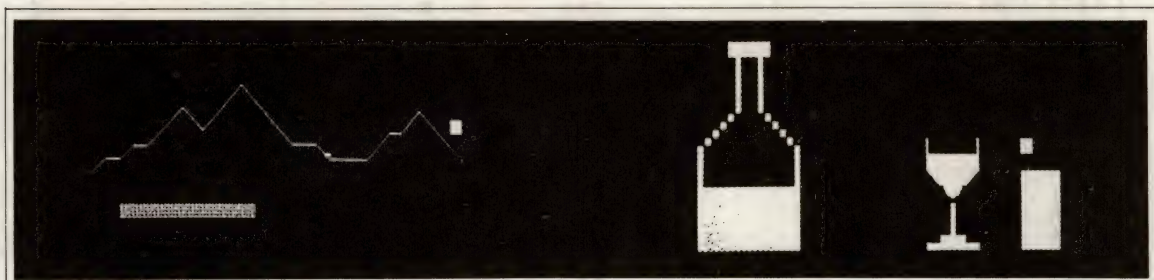
それから付録として全画面の消去。それからもう一つどうせメモリに余りがあるのですから，できた画面をしまっておく『画面退避』をやってよいのではないでしょうか……という機能をつけました。

遊び方

なぜ，遊び方なのか……マイコンとていっぱしのコンピュータ。何がしかの目的があってキー・ボードに向わなければ，これほど面白くない事はありません。

ですがこのプログラムでは，簡単に画面に絵が描けることから，目的がなくても，何とかヒマつぶしはできます。

さて，プログラムを入力したら当然8200からスタートです。これで画面はクリアされ，カーソルは左上にあります。

キーは次の三つのモードに対応してブロックに分かれます。

画面作図例

カーソル移動モード…RET, RUN, STORE, LOAD
制御モード……………ADRES, READ IN, READ
DEL, WRITE
キャラクタコード……0~Fのキー

そこで，上の四つのキーを操ってカーソルを任意の位置に運び，0~Fのキーにさわるとただちにキャラクター書込みモードになって，同時にLEDのキャラクタコードの部分が左にシフトされ，右はしに今入力したキーの値が入ります。キャラクタコードは16進2ケタですから，次に0~Fまで入力したとき，またLEDがシフトされ，ただちにそのときLEDに表示されているコードが書込まれ，次にカーソル移動なり，制御なり書込みの命令待ちになります。

難しくなりましたが，簡単にいうとカーソルを望みの位置に置いて，次に01というコード"A"を書きたければ，つづいて0，1と入力すればよろしい。このとき0のあとに0~F以外のキーを押しても無視されます。

このときLEDの1~4までにCURSOR X, Y, に対応して，1，2にXが3，4にY座標が表示されます。カーソルがすでにあるパターンに重なっても消えてしまうようなことはありませんからレイアウトを決める段階ではLEDの内容は気にしません。描いていって，完成した段階で，カーソルを動かして座標とコードを解析していくのです。

その絵をBASICを用いていかに描くかは，そのBASICの問題ですから，これを参考にしてプログラムを作っていきます。しかし，うまくBASICプログラムがかけなかったり，どこかでまちがっていたり，画面を消したあとだったら，もう一回調べるわけにはいきません。

そこで画面の退避をやるのです。私は8000~81FFまでにメモリがないのですが，是非ここにRAMを増設してダミーの画面をここに設けてください。

プログラムでは，私のシステムに合わせて，ここは9800~99FFにしてあります。画面が完成したと思ったら〔READ DEC〕キーを押しておきます。そうすればBASICのプログラム作成途中でもBREAKして8208からRUNさせ，〔WRITE〕キーを退避させておいた絵がよみがえります。そしてカーソルを動かして同じことをやるのです。

行のクリアというのは，カーソルの位置を含めてその右にあるものを消してしまいます。全クリアでは8200RUNと同じ状態に戻ります。

以上でなんとかUSSエンタープライズも描けそうな気がしてきました。

なお，BS関係で次のサブルーチンを用いています。

```
TVEX4    EQU   0FA6CH
TVEX6    EQU   0FAB9H
TVEX7    EQU   0FADCH
TRANS1   EQU   0F922H
TRANS2   EQU   0F937H
```

TV画面作図プログラム・リスト

アドレス	機械語	ラベル	ニーモニック	オペランド	備考
			ORG	8200H	
8200	CD6CFA	L1	CALL	FA6CH	CLEAR CRT
8203	3E20		MVI	A, 20	
8205	32007E		STA	7F00H	
8208	1E01	L2	MVI	E, 01	FIRST POSITION
820A	1601		MVI	D, 01	
820C	3E1D	L3	MVI	A, 1D	HEX-DESIMAL
820E	BB		CMP	E	
820F	D21B82		JNC	821BH	
8212	7B		MOV	A, E	
8213	C612		ADI	12	
8215	32F483		STA	83F4H	
8218	C33D82		JMP	823DH	
821B	3E13		MVI	A, 13	
821D	BB		CMP	E	
821E	D22A82		JNC	822AH	
8221	7B		MOV	A, E	
8222	C60C		ADI	0C	
8224	32F483		STA	83F4H	
8227	C33D82		JMP	823DH	
822A	3E09		MVI	A, 09	
822C	BB		CMP	E	

822D	D23982		JNC	8239H	
8230	7B		MOV	A, E	
8231	C606		ADI	06	
8233	32F483		STA	83F4H	
8236	C33D82		JMP	823DH	
8239	7B		MOV	A, E	
823A	32F483		STA	83F4H	
823D	7A		MOV	A, D	
823E	32F583		STA	83F5H	
8241	3E09		MVI	A, 09	
8243	BA		CMP	D	
8244	D24D82		JNC	824DH	
8247	7A		MOV	A, D	
8248	C606		ADI	06	
824A	32F583		STA	83F5H	
824D	62	L4	MOV	H, D	CAL. OF ADDRESS
824E	6B		MOV	L, E	
824F	227D84		SHLD	847DH	
8252	CDB9FA		CALL	FAB9H	
8255	2A7F84		LHLD	847FH	
8258	7E		MOV	A, M	
8259	32F783		STA	83F7H	
825C	E5	L5	PUSH	H	LED DISPLAY
825D	D5		PUSH	D	
825E	CDC001		CALL	01C0H	
8261	D1		POP	D	
8262	E1		POP	H	
8263	3E76		MVI	A, 76	
8265	32FD83		STA	83FDH	
8268	3E00		MVI	A, 00	
826A	32FC83		STA	83FCH	
826D	3E80	L6	MVI	A, 80	CURSOR
826F	4E		MOV	C, M	CHARACTER
8270	77		MOV	M, A	
8271	D5	L7	PUSH	D	KEYIN
8272	CD1602		CALL	0216H	
8275	D1		POP	D	
8276	71		MOV	M, C	
8277	FE17	L8	CPI	17	MOV CURS.
8279	C28682		JNZ	8286H	UNDER
827C	3E10		MVI	A, 10	
827E	BA		CMP	D	
827F	CA0C82		JZ	820CH	
8282	14		INR	D	
8283	C30C82		JMP	820CH	
8286	FE11		CPI	11	LEFT
8288	C29582		JNZ	8295H	
828B	3E01		MVI	A, 01	
828D	BB		CMP	E	
828E	CA0C82		JZ	820CH	
8291	1D		DCR	E	
8292	C30C82		JMP	820CH	
8295	FE10		CPI	10	RIGHT
8297	C2A482		JNZ	82A4H	
829A	3E20		MVI	A, 20	
829C	BB		CMP	E	
829D	CA0C82		JZ	820CH	
82A0	1C		INR	E	
82A1	C30C82		JMP	820CH	
82A4	FE16		CPI	16	UPPER
82A6	C2B382		JNZ	82B3H	
82A9	3E01		MVI	A, 01	
82AB	BA		CMP	D	
82AC	CA0C82		JZ	820CH	
82AF	15		DCR	D	
82B0	C30C82		JMP	820CH	
82B3	FE12	L9	CPI	12	ALL CLEAR
82B5	C2BB82		JNZ	82BBH	
82B8	C30082		JMP	8200H	
82BB	FE14	L10	CPI	14	LINE CLEAR
82BD	C2C682		JNZ	82C6H	
82C0	CDDCFA		CALL	FADCH	
82C3	C30C82		JMP	820CH	
82C6	FE13	L11	CPI	13	GAMEN TAIHI
82C8	C2E282		JNZ	82E2H	

アドレス	機 械 語	ラ ベ ル	ニーモニック	オペランド	備 考
82CB	D5		PUSH	D	
82CC	E5		PUSH	H	
82CD	C5		PUSH	B	
82CE	F5		PUSH	PSW	
82CF	11007E		LXI	D, 7E00	
82D2	21FF7F		LXI	H, 7FFF	
82D5	010098(010080)		LXI	B, 9800	(LXI B, 8000)
82D8	CD22F9		CALL	F922H	
82DB	F1		POP	PSW	
82DC	C1		POP	B	
82DD	E1		POP	H	
82DE	D1		POP	D	
82DF	C30C82		JMP	820CH	
82E2	FE15	L12	CPI	15	GAMEN HUKKI
82E4	C2FE82		JNZ	82FEH	
82E7	D5		PUSH	D	
82E8	C5		PUSH	B	
82E9	E5		PUSH	H	
82EA	F5		PUSH	PSW	
82EB	110098(110080)		LXI	D, 9800	(LXI D, 8000)
82EE	01FF99(01FF81)		LXI	B, 99FF	(LXI B, 81FF)
82F1	21007E		LXI	H, 7E00	
82F4	CD37F9		CALL	F937H	
82F7	F1		POP	PSW	
82F8	E1		POP	H	
82F9	C1		POP	B	
82FA	D1		POP	D.	
82FB	C30C82		JMP	820CH	
82FE	47	L13	MOV	B, A	KAKIKOMI
82FF	3E00		MVI	A, 00	
8301	325883		STA	8358H	
8304	D5		PUSH	D	
8305	78		MOV	A, B	
8306	C31F83		JMP	831FH	
8309	3A5883		LDA	8358H	
830C	3C		INR	A	
830D	325883		STA	8358H	
8310	FE02		CPI	02	
8312	CA4283		JZ	8342H	
8315	C5		PUSH	B	
8316	CD1602		CALL	0216H	
8319	C1		POP	B	
831A	FE10		CPI	10	
831C	D21583		JNC	8315H	
831F	47		MOV	B, A	
8320	AF		XRA	A	
8321	79		MOV	A, C	
8322	17		RAL		
8323	B7		ORA	A	
8324	17		RAL		
8325	B7		ORA	A	
8326	17		RAL		
8327	B7		ORA	A	
8328	17		RAL		
8329	80		ADD	B	
832A	32F783		STA	83F7H	
832D	4F		MOV	C, A	
832E	E5		PUSH	H	
832F	C5		PUSH	B	
8330	CDC001		CALL	01C0H	
8333	3E76		MVI	A, 76	
8335	32FD83		STA	83FDH	
8338	3E00		MVI	A, 00	
833A	32FC83		STA	83FCH	
833D	C1		POP	B	
833E	E1		POP	H	
833F	C30983		JMP	8309H	
8342	01F783		LXI	B, 83F7H	
8345	0A		LDAX	B	
8346	77		MOV	M, A	
8347	3E00		MVI	A, 00	
8349	325883		STA	8358H	
834C	CD1602		CALL	0216H	
834F	D1		POP	D	
8350	C37782		JMP	8277H	

26 マイコンの実践利用

家族計画プログラム

TRS-80

宮田利通

近年は都市の住宅事情が好ましからざる方向へ変化してきました。特に，大都会は良い例です。住宅問題に限らず，狭い国土，高い物価や教育費等で人々は苦しめられています。このようなことで，家族計画を意識して実行せざるをえないようになってきました。

この深刻な問題を解決する手段にマイクロコンピュータを使ってみたらどうでしょうか。簡単な方法で家族計画プログラムを作ることができます。ここではBASIC語によるプログラムを紹介します。ぜひ利用して快適な生活を楽しんでください。

荻野学説について

受胎調節といえば必ず引き合いに出される方法にオギノ式があります。この方法は，1924年に荻野久作博士が発表した学説に基づいています。この学説の主旨は，「婦人の排卵時期は月経周期に関係なく，予定月経前第16日から第12日に至る5日間である。」ということです。

一般にオギノ式はあまり効果がなく75%ぐらいであるといわれています。他の方法ではピル100%，コンドーム90%といわれています。しかしながらオギノ式の効果が低い原因はこの方法それ自体にあるのでなく，大部分はあやまって覚えていたためによる失敗なのです。この家族計画プログラムは荻野学説をプログラムしたものですが，正確なデータを得て，それを入力するだけでより高い効果を期待することができます。

プログラムの使用法について

プログラムの実行は，特別に難しいことはありません。ただし，正しく使用してもらうため最初に重要な概念を説明します。それは「月経周期」ということです。

月経周期とは，月経第一日目から次の月経のはじまる前日までの日数のことです。例えば4月1日から5日まで月経が続き，次の月経周期は29日となります。

この月経周期は，同じ婦人であっても不定期なので，長期間に渡って観察して最大月経周期と最小月経周期を知る必要があります。少なくとも1年間のデータを使うとプログラムの効果が期待できます。

さて，以上のことがわかったならば，ソースリストに従ってプログラムを正確に入力してください。RUNとタイプするとプログラムが実行され，**第26—1図**のようにディスプレイされます。

“WHAT IS YOUR MINIMUM MENSTRUATION?”とコンピュータが尋ねてくるので，最小月経周期を入力します。次に，“WHAT IS YOUR MAXIMUM MENSTRUATION?”に対して，最大月経期を入力します。“WHEN DID YOUR LATE MENSTRUATION BEGIN?”は，月経の始まった日付のことを尋ねています。

第26-2表は，最小月経周期28日，最大月経周期33日，月経開始日3月15日の例です。**第26-3表**は，その計算結果の出力例です。すなわち，妊娠の可能性のある期間は，3月24日～4月5日の間ということになります。

《第26-1表》プログラムを実行した直後の表示

```
%%%%  FAMILY  PRANING  PROGRAM  %%%%

WHAT IS YOUR MINIMUM MENSTRUATION?
```

《第26-2表》必要なデータを入力したときの表示

```
%%%%  FAMILY  PRANING  PROGRAM  %%%%

WHAT IS YOUR MINIMUM MENSTRUATION? 28
WHAT IS YOUR MAXIMUM MENSTRUATION? 33
WHEN DID YOUR LATE MENSTRUATION BEGIN ?
   > MONTH (NO ALPHABET, BUT FIGURE)? 3
  > MONHT (NO ALPHABET, BUT FIGURE)? 15
```

《第26-3表》妊娠可能期間の表示

```
* * * THE DAYS FOR PREGNACY * * *

MONTH  : 3
DAYS  : 24 25 26 27 28 29 30 31

MONTH  : 4
DAYS  : 1 2 3 4 5
```

プログラム・リスト

```
5 REM ** TRS-80 BASIC ** AUTHOR T.MIYATA **
7 POKE 16553,255
8 CLS
10 PRINT "%%%%  FAMILY  PRANING  PROGRAM  %%%%"
20 PRINT
30 INPUT "WHAT IS YOUR MINIMUM MENSTRUATION";PI
40 INPUT "WHAT IS YOUR MAXIMUM MENSTRUATION";PA
50 PRINT "WHEN DID YOUR LATE MENSTRUATION BEGIN ?"
60 INPUT"   > MONTH (NO ALPHABET, BUT FIGURE)";MN
70 INPUT"   > DAY   (NO ALPHABET, BUT FIGURE)";DY
80 S=DY+PI-19 : E=DY+PA-12
90 GOSUB 600
100 CLS
105 PRINT "* * *  THE  DAYS  FOR  PREGNANCY  * * *"
110 PRINT
120 T=0
130 FOR I=S TO E
140  R=I-T
150  IF R>H THEN T=H : GOTO 500
160  IF I=S THEN PRINT "MONTH : ";MN : PRINT "DAYS : ";
170  PRINT R;
180 NEXT I
190 PRINT : PRINT : PRINT : PRINT
200 GOTO 10
500 REM ------------------------------------------
505 MN=MN+1
510 IF MN=13 THEN MN=1
520 IF I=S THEN 540
530 PRINT : PRINT
540 PRINT "MONTH : ";MN : PRINT "DAYS : ";
550 R=I-T
560 GOSUB 600
570 GOTO 170
600 REM ----------- SUB ----------------------
605 FOR J=1 TO MN
610  READ H
620  DATA 31,28,31,30,31,30,31,31,30,31,30,31
630 NEXT J
640 RESTORE
650 RETURN
660 END
```

27

電気回路の自動設計

定電圧回路の設計

H68/TRL I

宮田利通

マイクロコンピュータの応用の一例として，定電圧回路を設計するプログラムです。

このプログラムは，必要なデータ（例えば定電圧回路の入力電圧，出力電圧，負荷電流，トランジスタ定電圧ダイオードの各定数や電圧，電流）をキーボードより順次入力するだけで，回路定数やトランジスタのコレクタ損失，コレクタ電流をCRT上に表示するようになっています。

又，プログラムを記述するために用いた言語は，日立製のBASIC-Iです。他社のBASICを使用している人は，多少の変更をしてください。

BASIC-Iについて

ここで使用しているBASICについて簡単に説明します。もしも，変更を必要とする場合は参考にして下さい。

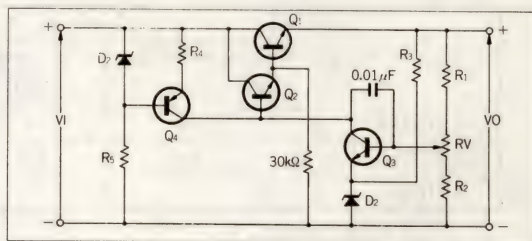
キーワードの省略形について述べますと，プログラム中のすべてのLETを省略することができるので，ここでは使用していません。入出文については，PRINTをPR, INPUTをINとそれぞれ省略して使いました。FOR～NEXTでは，NEXTをNEXと省略しました。

そのほかに注意することとしては，PRINT文中のCHR$ ($0C)は，画面を一度クリアーしてからカーソルをホームポジションへ移します。「.」は，次の出力をすぐ後に続け，「；」はスペースを1個設けることを表します。同一文中で「：」を使うと複数の文を記述することができます。

このインタープリタであつかえる数値は，－32768から＋32767の間の整数です。実際に入力するデータ（トランジスタのベース・エミッタ間電圧，定電圧ダイオードのツェナー電圧）は小数を含むものがあるので，それらの電圧値を10倍あるいは100倍して整数に変更後，入力するようにしました。

プログラムの実行例

H68/TR＋TVインターフェースを持っている場合は，166ページに掲げたソースプログラムをそのまま入力します。入力してRUNコマンドを実行すると，CRT上にTHE CIRCUIT DESING OF VOLTAGE REGULATORの文字が表示され，プログラムが実行されたことがわかります。

CRTにデータの入力要求のメッセージが次々に現れるので，必要とする数値を入力します。入力が正しく行われると画面がクリアーされ，回路の各抵抗値が計算され出力されます。又，制御回

《第27-1図》定電圧回路の基本形

路で使うトランジスタの制限値も同じ画面上に表示されるので，規格表の最大定格と比較することができます。

具体的に数値をあげながら，回路設計の方法を手順を追って説明します。このプログラムが適用される定電圧回路を**第27―1図**に示します。

(1)仕様の決定

まず第一に決めることは，設計する回路の仕様です。入力電圧，出力電圧，負荷電流の三つを決めます。各値を**第27―1表**に示します。

(2)トランジスタの決定

トランジスタは，規格表や価格表を参考にして決めます。この時，留意する点は最大定格です。コレクタ電流，コレクタ損失，コレクタ・エミッタ間電圧などの定格に特に注意します。最大定格が同じでも価格の高いもの安いもの，入手し難いものし易いものがあるので，近くのパーツ店で入手可能でありかつ安価のトランジスタを選ぶのが良いでしょう。

回路図中，Q_1, Q_2, Q_3はNPN形トランジスタですが，Q_4はPNP形トランジスタなので間違えないようにします。

以上の点を考慮して**第27―2表**のようにトランジスタを選びました。

(3)定電圧ダイオードの決定

定電圧ダイオードのツェナー電圧は，定電圧回路の入出力電圧より小さい値を選ばなければなりません。もし，大き目の値を使うと，抵抗R_1, R_3, R_5の計算値が負数になってしまい設計不能になります。

第27―3表に定電圧ダイオードの特性を示します。

(4)データの入力

マイクロコンピュータは，入力データを順次要求してくるので，次のように数値をキーインする。

第27―1表の値について，MAXIMUM UNREGULATED INPUT VOLTAGE 14, MINIMUM UNREGULATED INPUT VOLTAGE 10, REGULATED OUTPUT VOLTAGE 9, LOAD CURRENT〔MA〕500と入力します。

第27―2表よりトランジスタのデータを入力します。CURRENT GAIN OF Q_1 25, CURRENT GAIN OF Q_2 350, VOLTAGE (B―E) OF Q_3〔V〕×100 67 (0.67を100倍した値)，CURRENT GAIN OF Q_4 80, VOTAGE (B―E) OF Q_4〔V〕×100 20 (0.20を100倍した値) となる。

定電圧ダイオードは，D_1とD_2に同じものを使うので，**第27―3表**より，ZENER VOLTAGE OF D_1〔V〕×10 56 (5.6を10倍した値)，ZENER CURRENT OF D_1〔MA〕10, ZENER VOLTAGE OF D_2〔V〕×10 56, ZENER CURRENT OF D_2〔MA〕10と入力します。

以上で設計プログラムを実行するのに必要なデータを全部入力したことになります。最後のデータを入れ終わると，画面がクリアーされ，計算結果が出力されます。

入力電圧	10～14 〔V〕
出力電圧	9 〔V〕
負荷電流	0～500 〔mA〕

《第27-1表》定電圧回路の仕様

02Z5.6A	
ツェナー電圧	5.6 〔V〕
ツェナー電流	10 〔mA〕

《第27-3表》定電圧ダイオード

トランジスタ	記号	最大定格			直流電流増幅率	ベースエミッタ間電圧〔V〕
		コレクター損失〔W〕	コレクタ電流〔A〕	コレクタベース間電圧〔V〕		
2SC1173	Q_1	10	3	30	25	―
2SC374	Q_2, Q_3	0.2	0.1	30	350	0.67
2SB54	Q_4	―	―	―	80	0.20

(注) 直流電流増幅率は、最小値をとる。

《第27-2表》トランジスタの規格

(5)出力結果

CRTには，定電圧回路の抵抗値と制御回路のトランジスタの制限値が表示されます。もし，入力データ中に不適当なデータが含まれていると，プログラムの実行は途中で中断され，エラーメッセージが出力されます。

マイクロコンピュータで示された抵抗値をそのまま採用するのは，賢明な方法ではありません。なぜなら，定電圧回路の組立に使用する抵抗は，市販されている値でなければならないからです。**第27—4表**に，出力結果と実際の抵抗値を示します。

トランジスタの制限値は次のようになります。Q_1の場合，コレクタ損失2500〔mW〕，コレクタ電流500〔mA〕，コレクタ・エミッタ間電圧5〔V〕となります。Q_2は，コレクタ損失95〔mW〕，コレクタ電流19〔mA〕，コレクタ・エミッタ間電圧5〔V〕となります。これらの各値と**第27—2表**の最大定格を比較して，すべて定格内であることを確認します。

(6)エラーメッセージ

入力データが不適当であった場合，＊ERROR＊の文字と共に不適当なデータの関係が表示されます。エラーメッセージに使われる記号について説明しましょう。

VI, VOは，それぞれ定電圧回路の入力電圧，出力電圧を表わします。V (B−E) 3とV (B−E) 4は，トランジスタQ_3とQ_4のベース・エミック間電圧です。又，VZ_1とVZ_2は，定電圧ダイオードD_1とD_2のツェナー電圧を示します。

抵抗器	出力結果〔OHMS〕	実際の抵抗値〔Ω〕
R_1	546	650
R_2	754	750
RV	500	500
R_3	340	360
R_4	5400	5.6〔kΩ〕
R_5	440	470

《第27-4表》定電圧回路の抵抗

例をあげて説明すると，**第27—1表**の仕様を変更して，出力電圧を5〔V〕と設定してみます。各データを入力すると

＊ERROR＊

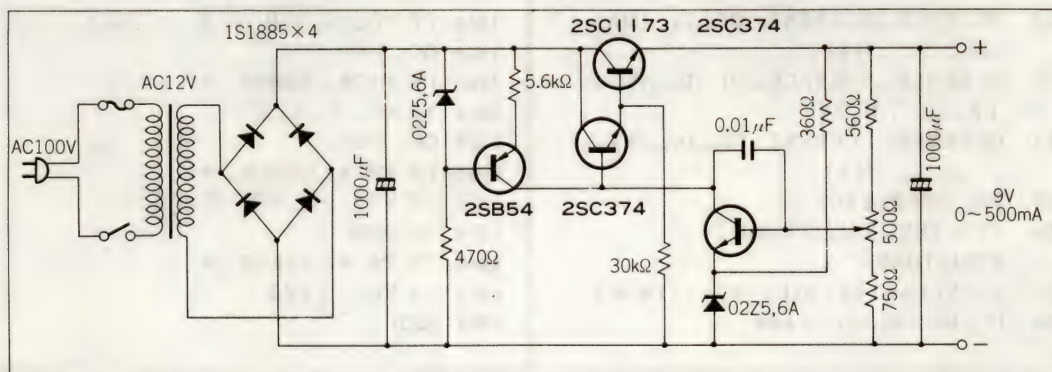
VO<V(B−E)3＋VZ1

のようにエラーメッセージが出力されます。すなわち，出力電圧5〔V〕，Q_3のベース・エミッタ間電圧0.67〔V〕，ツェナー電圧5.6〔V〕であるから，5〔V〕<0.67〔V〕＋5.6〔V〕となり，設計不能であることを示します。

(7)定電圧回路

出力結果をもとに，**第27—1図**の回路に抵抗値を書き入れればよいのです。**第27—2図**に実際の回路図を示します。図中の0.01〔μF〕のコンデンサと30〔kΩ〕の抵抗は，回路動作を安定にするために入れてあります。

この定電圧回路設計プログラムは，トランジスタが決まれば，簡単に設計できるし，色々と値を入力して目的にあった回路を設計することができます。

このプログラムをコーディングするとき用いた

《第27-2図》プログラムにより完成した回路

設計式は，J.A.Walston, J. R. Miller : Transitonr Circuit Design, McGraw tlill pp・145—167です。

定電圧回路の詳しい設計を知りたい場合は，上記文献を参照することをお勧めします。

定電圧回路設計プログラム・リスト

```
1ø   DIM␣V(3), I(3), H(3), B(2), D(2)
15   FOR␣L=1␣TO␣32 : PR"=". : NXT␣L
2ø   PR"␣␣␣THE␣CIRCUIT␣DESIGN␣OF"
25   PR"␣␣␣VOLTAGE␣REGULATOR"
3ø   FOR␣L= 1 ␣TO␣32 : PR"=". : NXT␣L
35   PR
4ø   PR"***␣SPECIFICATION␣***"
      : PR
45   PR"MAXIMUM␣UNREGULATED␣
     INPUT"
5ø   IN"VOLTAGE␣[V]␣␣␣", V(1)
55   PR"MINIMUM␣UNREGULATED␣
     INPUT"
6ø   IN"VOTAGE␣[V]␣␣␣", V(2)
65   PR"REGULATED␣OUTPUT␣
     VOLTAGE␣[V]"
7ø   IN␣V(3)
75   IN"LOAD␣CURRENT␣[MA]␣␣␣",
     I(3)
8ø   PR
85   PR"***␣DATA␣OF␣TRANSISTORS␣
     ***"
9ø   IN"CURRENT␣GAIN␣OF␣Q1␣␣␣",
     H(1)
95   IN"CURRENT␣GAIN␣OF␣Q2␣␣␣",
     H(2)
1øø  IN"VOLTAGE(B-E)␣OF␣Q3␣[V]×1øø
     ␣␣␣", B(1)
1ø5  IN"CURRENT␣GAIN␣OF␣Q4␣␣␣",
     H(3)
11ø  IN"VOLTAGE(B-E)␣OF␣Q4␣[V]× 1øø
     ␣␣␣, B(2)
115  PR
12ø  PR"***␣DATA␣OF␣ZENER␣
     DIODES␣***"
125  IN"ZENER␣VOLTAGE␣OF␣D1␣[V]×
     1ø␣␣␣", D(1)
13ø  IN"ZENER␣CURRENT␣OF␣D1␣[MA]
     ␣␣␣␣␣", I(1)
135  IN"ZENER␣VOLTAGE␣OF␣D2␣[V]×
     1ø␣␣␣", D(2)
14ø  IN"ZENER␣CURRENT␣OF␣D2␣[MA]
     ␣␣␣␣␣", I(2)
145  PR␣CHR$($øC).
15ø  PR">THE␣VALUES␣OF␣
     RESISTORS<"
155  R=(V(3)*1øø-B(1)-D(1)*1ø)*2
16ø  IF␣R<=ø␣GO␣1øøø
165  PR"R1=". R; "[OHMS]",
17ø  R=(B(1)+D(1)*1ø)*2-5øø
175  PR"R2=".R; "[OHMS]",
18ø  PR"RV=5øø␣[OHMS]",
185  X=1øø/I(1)
19ø  R=(V(3)*1ø-D(1)*X
195  IF␣R<=ø␣GO␣2øøø
2øø  PR"R3=".R; "[OHMS]",
2ø5  Y=I(3)*4/(H(1)+1)/(H(2)+1)
21ø  IF␣Y=ø␣Y=1
215  X=Y+Y/H(3)
22ø  R=(D(2)*1ø-B(2))*1ø/X
225  IF␣R<=ø␣GO␣3øøø
23ø  PR"R4=".R; "[OHMS]",
235  X=1øø/(I(2)+Y/H(3))
24ø  R=(V(2)*1ø-D(2))*X
245  IF␣R<=ø␣GO␣4øøø
25ø  PR"R5=".R; "[OHMS]"
255  PR
26ø  PR">THE␣LIMITATIONS␣OF␣
     TRANSIS␣TORS<".
265  PR"Q1 :"
27ø  J=V(1)-V(3)
275  P=I(3)*J
28ø  PR"DISSIPATION=".P; "[MW]"
285  PR"COLLECTOR␣CURRENT=".I(3);
     "[MA]"
29ø  PR"VOLTAGE(C-E)=".J; "[V]"
295  PR"Q2 :"
3øø  K=I(3)/(H(1)+1)
3ø5  P=K*J
31ø  PR"DISSIPATION=".P; "[MW]"
315  PR"COLLECTOR␣CURRENT=".K;
     "[MA]"
32ø  PR"VOLTAGE(C-E)=".J; "[V]"
325  GO␣9999
1øøø PR: PR"*␣ERROR␣*"
1ø1ø PR"VO␣<␣V(B-E) 3␣+␣VZ1"
1ø2ø GO␣9999
2øøø PR; PR"*␣ERROR␣*"
2ø1ø PR"VO␣<␣VZ1"
2ø2ø GO␣9999
3øøø PR: PR"*␣ERROR␣*"
3ø1ø PR"VZ2␣<␣V(B-E) 4"
3ø2ø GO␣9999
4øøø PR: PR"*␣ERROR␣*"
4ø1ø PR"VI␣<␣VZ2"
9999 END
```

28

教育分野への応用

一次方程式の解法と学習

TK-80BSL I

天野卓夫

このプログラムは，中学校の2年生で学習することになっている一次方程式の問題作成についてです。プログラムの内容は，最も基本的なタイプ**AX+B=C**の問題に限定してみました。なお，使用機種は，NECのTK-80と80BSを使用しています。

構 想 1

まず，乱数関数RNDを使用して，問題の式をつくります。この場合，TK-80BS・LEVEL I・BASICでは小数点が計算できないということから，求める解Xが整数値である必要があります。そこで逆手に考えて，まずAとB，そしてXの値を決定します。それから**C=AX+B**としてCを求めます。

なお，プログラム上では解を，XではなくYとしていますが，これは後で判定の際のためのトリックです。

構 想2

AX+B=CのA，B，Cが正数に限られたのでは，あまり実用性に乏しいので，場合によって，負数であることも必要と考え，再びRNDを使って次のように処理します。

```
E=RND(10);IFE<6  A=-A
```

なお，負の数が参加することから，式の表示をととのえる上で多少手を加えています（行番号85)。

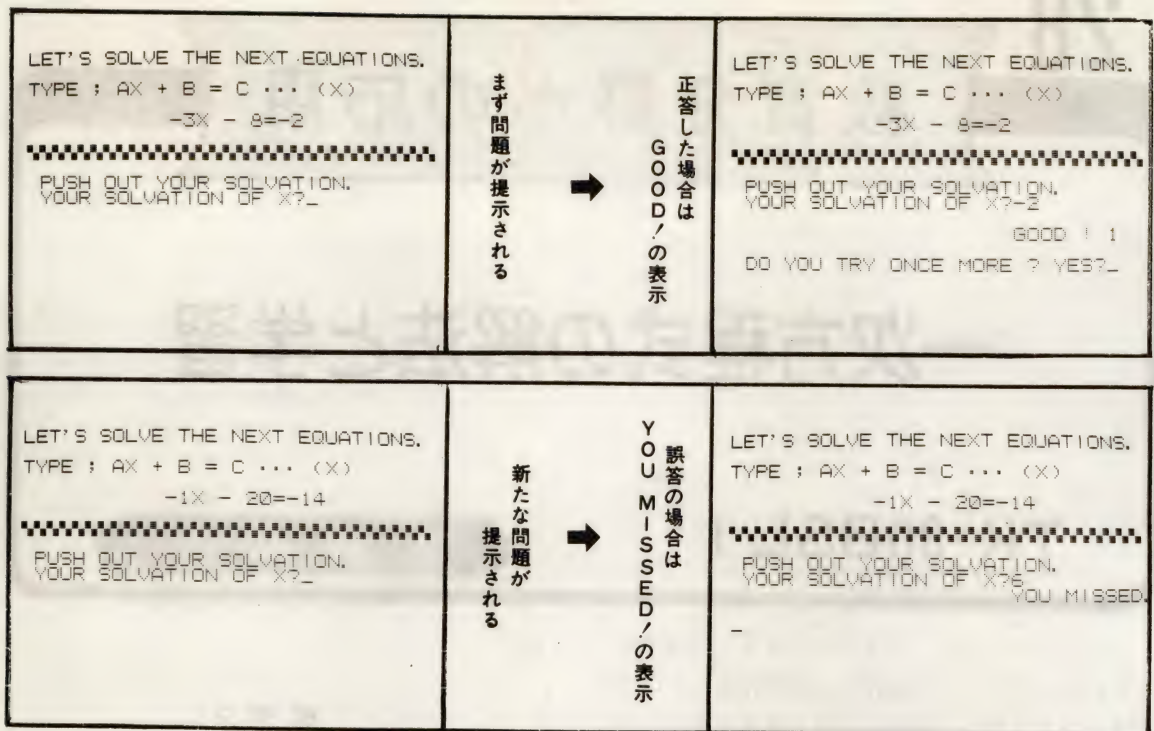
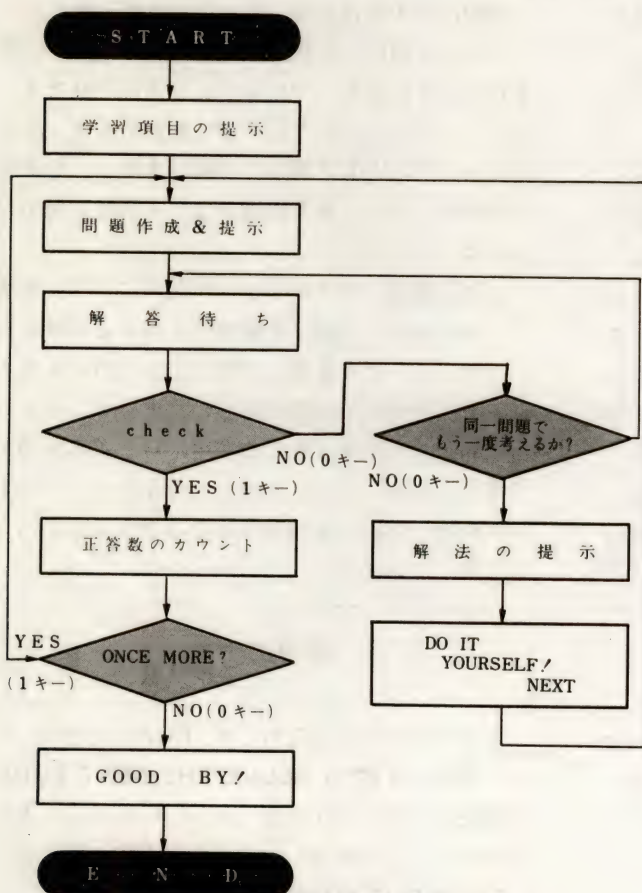
構 想3

問題に対する答えが，あっているか，まちがっているかを判断した後，あっていたら「もう一度するか」それとも「やめるか」を選択させます。もし，まちがっていたら「同一問題でもう一度考えるか」それとも「解き方を提示するか」を選択させます。但し，正答の場合は，正答数を提示できるようにします。

また，実際，近所の中学生が遊びにきて，実験してみた結果，スピード調整等を加える必要があり，サブルーチンを加えて解法する人の考えるスピードに合わせて，時間待ちの処理をところどころに加えています。その他多少，いたずらも加えてありますが，けっこう楽しく行なえるようです。いずれは，学校へも持ちこんでみようなどといろいろ考えています。

学習の方法

まずプログラムを入力して，RUNさせます。すると画面に**"LET'S SOLVE THE NEXT EQUATIONS."**と表示され問題（AX+B=Cの形：Xを未知数として求める）が提示されます。そして，

```
PUSH OUT YOUR SOLVATION
          YOUR SOLVATION OF X?
```

学習の進行

《第28―1図》一次方程式解法フローチャート

と聞いてきます。「いくらになりましたか？」という問に対して学習者は答えと思う数字のキーを押します。数値を入力して復改キーを押し，もし，その答えが正しい場合,

“GOOD！”

の点滅をして，正答数がカウントされGOODの表示の横に表示されます。そうして,

DO YOU TRY ONCE MORE?　YES

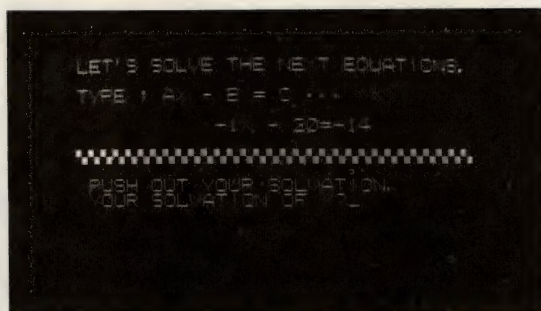
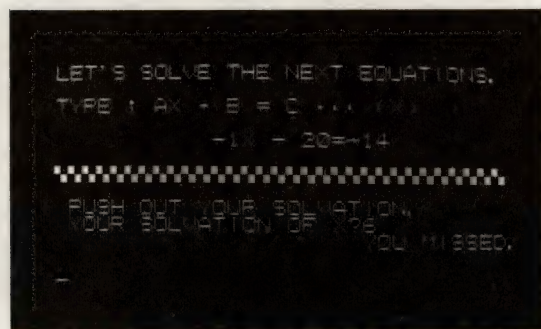
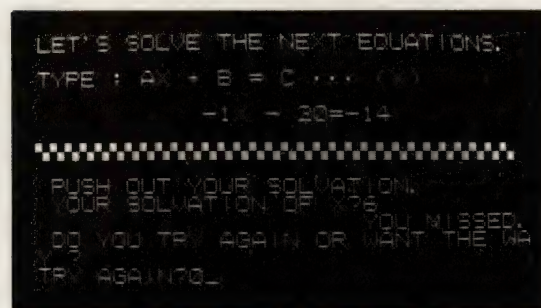
―?と聞いてきます。「もう一度しますか？」の問に対して, YESの場合は1, NOの場合は0を押します。さらに問題が提示されます。学習者の答えが違っている場合は,

YOU MISSED

DO YOU TRY AGAIN OR WANT

THE WAY?

TRY AGAIN?―と聞いてきます。

「同一問題でもう一度考えますか，それとも解法を聞きますか？」の問に対して，もう一度同一問題でやる場合は1を入力します。すると最初に問題が提示され

STEP1

STEP2

STEP3

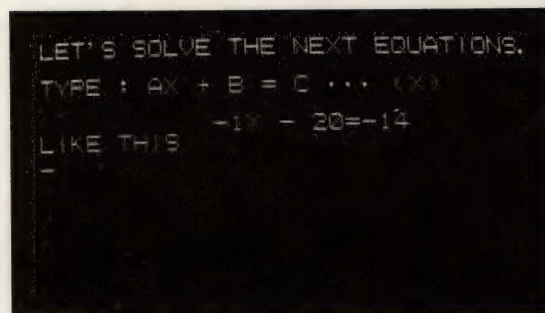
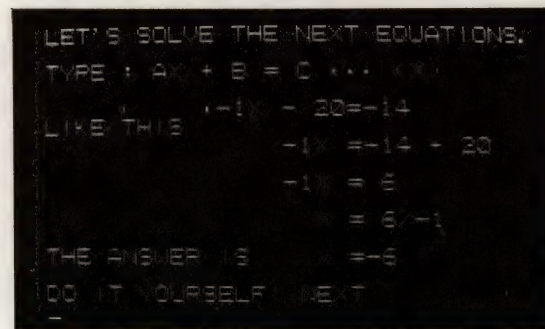
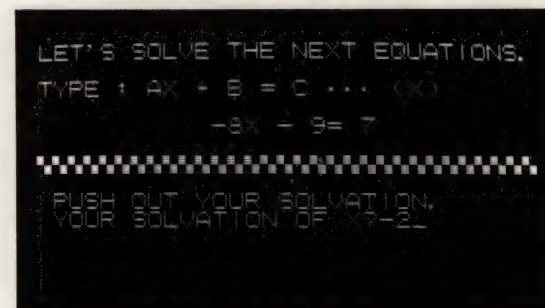
た状態にもどります。何度でもまちがえるようなら,学習者は0キーを入力して解法を聞くことができます。たとえば問題が　−8X−9=7だとすると

LIKE THIS

$$-8X=7+9$$

$$-8X=16$$

$$X=16/-8$$

THE ANSWER IS X=−2

STEP4

STEP5

STEP6

DO IT YOURSELF! NEXTと表示され，解法を提示して，新しい問題を提示してくれます。

学習を終了したい場合は，正答して，

DO YOU TRY ONCE MORE?の提示のあとに0キーを押せば終了です。画面上をGOOD BY!が上下に移動して，画面をCLEARして入力待ちの状態になります。

```
   10 CL.
   20 P."LET'S SOLVE THE NEXT EQ
UATIONS."
   30 P.;
   35 Q=0
   40 P."TYPE ; AX + B = C ··· (
X)"
   45 GOS.2000
   50 A=RND(10),B=RND(20),Y=RND(
10)
   53 IF ((A=1)*(B=1)) G.50
   55 IF ((A=1)*(B=11)) G.50
   60 E=RND(10);IF E<6 A=-A;F=RN
D(10);IF F<6 B=-B;G=RND(10);IF G
<6Y=-Y
   70 C=A*Y+B
   73 IF B>0 Z=1
   75 IF B<0 Z=0 ;G.83
   80 CU.5,5;P.A,"X +",#1,B,"=",
C
   81 G.88
   83 B=-B
   85 CU.5,5;P.A,"X -",#1,B,"=",
C
   88 GOS.2000
   90 FOR I=1 TO 31
   91 J=7
   92 CU.I,J
   93 FOR T=1 TO 50
   94 N.T
   95 P.HA2
   96 N.I
   97 CU.2,9
   98 P."PUSH OUT YOUR SOLVATION
 "
  100 P.;
  105 CU.2,10
  110 IN."YOUR SOLVATION OF ",X
  111 FOR G=1 TO 500
  112 N.G
  120 IF X#Y G.200
  127 FOR T=1 TO 7
  128 CU.22,12
  130 P."GOOD !";GOS.2300
  132 CU.22,12;P.;GOS.2300
  133 N.T
  134 CU.22,12;P."GOOD !"
  140 Q=Q+1;CU.28,12;P.#1,Q
  145 G.249
  200 CU.22,11
  210 P."YOU MISSED.";G.900
  249 CU.2,14
  250 P."DO YOU TRY ONCE MORE ?"

  251 CU.25,14
  252 IN."YE",S
  254 IF ((S#0)*(S#1)) G.249
  255 IF S=0 G.401
  256 IF S=1 G.257
  257 GOS.2000
  260 CU.2,9;P.;CU.2,10;P.;
  261 CU.22,12;P.;
  262 CU.2,14;P.;
  263 P.;
  265 GOS.2000
  270 CU.1,7;P.;
  315 CU.5,5;P.;
  320 GOS.2100
  350 G.50
  401 GOS.2100
  403 C.
  410 I=14
  420 FOR J=1 TO 16
  430 CU.I,J;P."GOOD BY !";GOS.2
300
  431 CU.I,J;P.;
  440 N.J
  450 GOS.2200
  500 C.;S.
  900 GOS.2100
 1000 CU.2,12;P."DO YOU TRY AGAI
N OR WANT THE WAY ?"
 1020 IN."TRY AGAI",N
 1030 IF N=1 G.1040
 1035 IF N=0 G.1100
 1040 FOR G=1 TO 200
 1045 N.G
 1050 CU.2,12;P.;P.;P.;
 1060 P.;
 1065 CU.22,11;P.;
 1070 G.88
 1100 GOS.2200
 1120 CU.2,9;P.;CU.2,10;P.;
 1130 CU.2,12;P.;P.;P.;
 1131 CU.1,7;P.;CU.22,11;P.;
 1140 CU.1,6;P."LIKE THIS"
 1145 GOS.2200
 1149 IF Z=1 G.1160
 1150 IF Z=0 G.1200
 1160 CU.9,7;P.A,#1,"X =",C," -"
,B
 1170 G.1250
 1200 CU.9,7;P.A,#1,"X =",C," +"
,B
 1210 G.1260
 1250 CU.9,9
 1251 GOS.2200
 1255 P=C-B;G.1300
 1260 CU.9,9
 1261 GOS.2200
 1265 P=C+B;G.1300
 1300 P.A,#1,"X =",P
 1310 GOS.2200
 1320 CU.18,11;P."X =",#1,P,"/",
A
 1335 L=P/A
 1336 GOS.2200
 1340 CU.1,13;P."THE ANSWER IS
 X =",#1,L
 1345 GOS.2200
 1346 GOS.2200
 1350 P.;P."DO IT YOURSELF! NEXT
"
 1360 GOS.2200
 1361 GOS.2200
 1370 CU.1,5;P.;P.;P.;P.;P.;P.;P
.;P.;P.;P.;P.;
 1375 G.50
 2000 FOR G=1 TO 1000
 2010 N.G;R.
 2100 FOR G=1 TO 2000
 2110 N.G;R.
 2200 FOR G=1 TO 3000
 2210 N.G;R.
 2300 FOR G=1 TO 30
 2310 N.G;R.
>_
```

29 教育事務処理への利用

改良版
成績処理プログラム

TK-80BSL II

山県昌彦

学校のクラスと各人の成績を入力して順位，平均点，標準偏差などを計算して画面に表示するプログラムです。

入力

画面に〝○バン　ナンテン？〞と1番から順次点数を聞いてきますから，番号順に点数を入力します。

欠席者は999，入力ミスの場合は－999を入れてから正しい点を入れ直し，全部入れ終わったら－100を入れます（**写真29—1，2**）。

出力

(1)在籍数，実受験者数，平均点数，標準偏差と0～9，10～19，…，90～99の10点幅と100点の度数分布ヒストグラムを表示します。その際，一つの棒グラフは20人までは，細い帯，20人を超えるとその分だけ太い帯で表すようにしてあります（**写真29—3**）。

(2)成績順に，順位，点数，番号を表示します。

```
 バンゴウ ジュン ニ テンスウ ヲ イレル、タダシ
ケッセキ ハ 999，イレマチガエタラ -999 ヲ イレテ
カラ イレナオス。 ゼンブイレタラ -100 ヲオセ.
(1) ヘイキン.ヒョウジュンヘンサ.ドスウ ブンプ
(2) セイセキ ジュン: テン.バンゴウ
(3) バンゴウ ジュン: テン.ジュンイ.ヘンサチ
<C.> ニ ヨリ (1)-->(2)-->(3) ト ススム.
 1 バン ナンテン?70
 2 バン ナンテン?_
```

《写真29—1》得点の入力(1)

```
<C.> ニ ヨリ (1)-->(2)-->(3) ト ススム.
 1 バン ナンテン?70
 2 バン ナンテン?85
 3 バン ナンテン?75
 4 バン ナンテン?100
 5 バン ナンテン?999
 6 バン ナンテン?90
 7 バン ナンテン?85
 8 バン ナンテン?50
 9 バン ナンテン?-999
 8 バン ナンテン?70
 9 バン ナンテン?100
 10 バン ナンテン?999
 11 バン ナンテン?-100_
```

《写真29—2》得点の入力(2)

《写真29—3》ヒストグラム

《写真29—4》成績順の表示

```
 1  バン 70  テン 7  イ  S= 37
 2  バン 85  テン 4  イ  S= 50.5
 3  バン 75  テン 6  イ  S= 41.5
 4  バン 100 テン 1  イ  S= 64.1
 5  バン ケッセキ
 6  バン 90  テン 3  イ  S= 55
 7  バン 85  テン 4  イ  S= 50.5
 8  バン 70  テン 7  イ  S= 37
 9  バン 100 テン 1  イ  S= 64.1
10  バン ケッセキ >_
```

《写真29—5》出席番号順による表示(1)

同点者は，同順位にし，また記録のために15人ずつで画面がストップし，Ｃ．で次の15人を表示するようにしました **（写真29—4）**。

(3)実際の成績処理では，(2)をもう一度，番号順に並べ直した方が便利です。それで今度は番号順に，番号，点数，順位，偏差値と15人ずつ表示するようにしました。その場合，欠席者は **〝○バンケッセキ〟** と表示するようにしてあります **（写真29—5，6）**。

（註） 1．(1)→(2)→(3)はＣ．(Continue)によって移ります。

2．平均点　M

$=\Sigma Xi/N$

標準偏差　S

$=\sqrt{\Sigma Xi^2/N-M^2}$

偏差値　D(Xi)

$=50+10(Xi-M)/S$

N：総人数

を用いています。

```
 1  バン 70  テン 7  イ  S= 37
 2  バン 85  テン 4  イ  S= 50.5
 3  バン 75  テン 6  イ  S= 41.5
 4  バン 100 テン 1  イ  S= 64.1
 5  バン ケッセキ
 6  バン 90  テン 3  イ  S= 55
 7  バン 85  テン 4  イ  S= 50.5
 8  バン 70  テン 7  イ  S= 37
 9  バン 100 テン 1  イ  S= 64.1
10  バン ケッセキ
11  バン ケッセキ
12  バン ケッセキ
13  バン ケッセキ
14  バン ケッセキ >_
```

《写真29—6》出席番号順による表示(2)

操作上の注意

入力された点数は，配列を用いて点数を整数部に番号を小数点第1，2位を用いて，たとえば，2番80点は，A(2)＝80.02と作って貯蔵しておきます。

欠席者は飛ばしていきますから，配列の番号(A(X)のX）は必ずしも生徒の番号とは一致しません。

出力の(2)では，A(X)の整数部の比較をして，その大小の順に内容を入れかえます。たとえば，上記の80点が順位4番なら，A(4)＝80,0204と小数点第5,6位に順位を加えます。このとき，TK-80BSの場合では，有効数字6桁を超えると丸められてしまうので100点の得点の場合は整数部を0にしておきます。

出力(3)の段階では，整数部と小数点1，2位部とを入れかえ，たとえば，上記2番80点順位4のA(4)＝80.0204はA(4)＝2.8004に直して，今度は整数部(つまり番号）を比較して，小さい順に表示させるわけです。

その時，点数部が0のものについて，100点なのか本当の0点なのかを判別し，また番号が飛ぶところを探して，ケッセキと表示します。

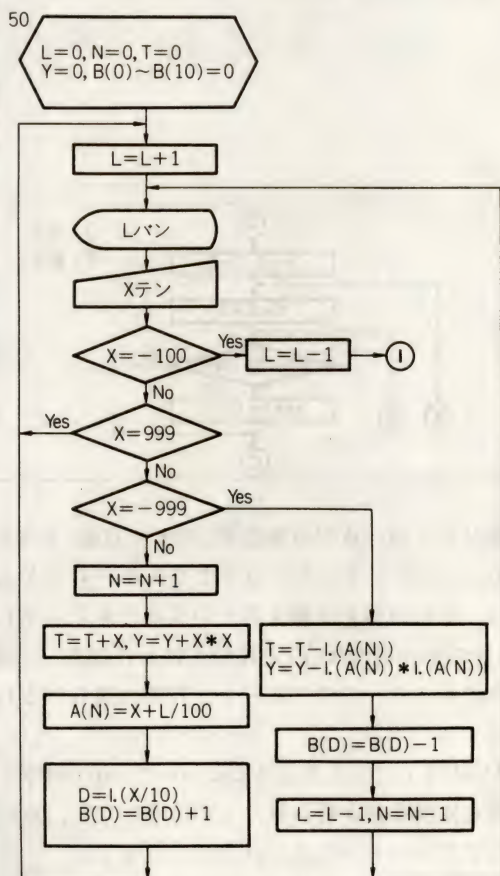
上の説明からおわかりのように6桁の数で各人の点数，番号，順位を表わすため人数は99人までに限られます。

100人以上の場合は成績順位の決定を(1)の各A(I)について他のすべてのA(J)（J≠1）と比較し，自分より大きいA(J)の個数を数えさせる方法によってもできますが，時間がかかってしまうでしょう。

各段階のフローチャートと　プログラムの説明

タイトル部

```
   5 CLEAR
  10 PRINT "       *** セ イ セ キ  シ
ョ リ ***"
  15 PRINT " ": PRINT " バンゴウ .
ジュン ニ テンスウ ヲ イレルノ タダシ"
  20 PRINT " ": PRINT "ケッセキ ハ 9
99,イレマチガエタラ -999 ヲ イレテ"
  25 PRINT " ": PRINT "カラ イレナオス
. ゼンブイレタラ -100 ヲオセ."
  30 PRINT " ": PRINT "(1) ヘイキン
.ヒョウジュンヘンサ.ドスウ ブンプ"
  35 PRINT " ": PRINT "(2) セイセキ
 ジュン: テン.バンゴウ"
  40 PRINT " ": PRINT "(3) バンコ
ウ ジュン: テン.ジュンイ. ヘンサチ"
```

```
(点数入力) L=番号(最終では在籍数), N=実受験者
          T=ΣXi, Y=ΣXi², A(N)=X+L/100
          B(D)=(10D≦X<10(D+1)の人数)
          B(10)=(X=100の人数)

50
L=0, N=0, T=0
Y=0, B(0)~B(10)=0
L=L+1
Lバン
Xテン
X=-100   Yes → L=L-1 → ①
No
X=999    Yes
No
X=-999   Yes
No
N=N+1                         T=T-I.(A(N))
                              Y=Y-I.(A(N))*I.(A(N))
T=T+X, Y=Y+X*X
A(N)=X+L/100                  B(D)=B(D)-1
D=I.(X/10)
B(D)=B(D)+1                   L=L-1, N=N-1
```

```
  50 LET L=0,N=0,T=0,Y=0
  55 DIM A(99),B(10)
  60 FOR I=0 TO 10
  65 LET B(I)=0: NEXT I
  70 LET L=L+1
  75 PRINT L;"バン ";
  80 INPUT "ナンテン"X
  85 IF X=-100 THEN  LET L=L-1:
GOTO 200
  90 IF X=999 THEN  GOTO 70
  95 IF X=-999 THEN  GOTO 150
 100 LET N=N+1
 105 LET T=T+X,Y=Y+X*X
 107 POKE 8624H,0H
 110 LET A(N)=X+L/100
 115 LET D=INT(X/10),B(D)=B(D)+
1: GOTO 70
 150 LET T=T-INT(A(N)),Y=Y-INT(
A(N))*INT(A(N))
 160 LET B(D)=B(D)-1
 170 LET L=L-1,N=N-1: GOTO 75
```

これにより，たとえば

番号	点数	
1	70	
2	80	
3	999	
4	70	
5	100	を入力すると

A(1)=70.01

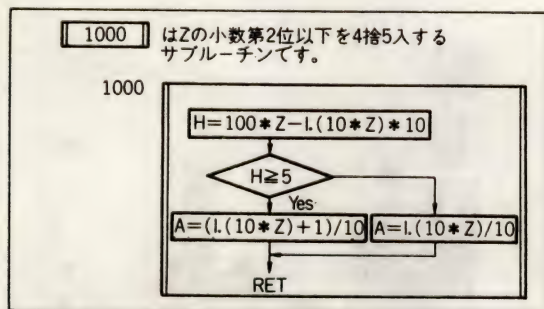
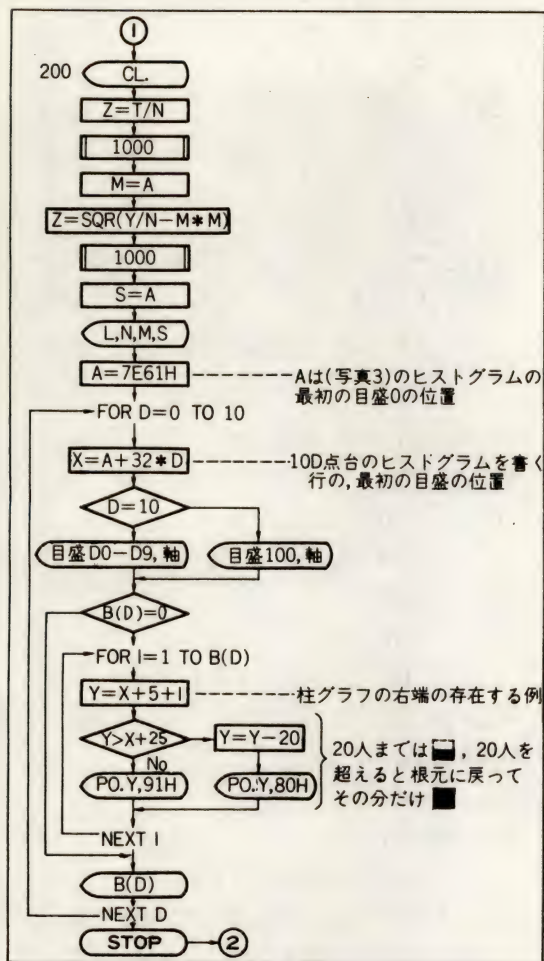
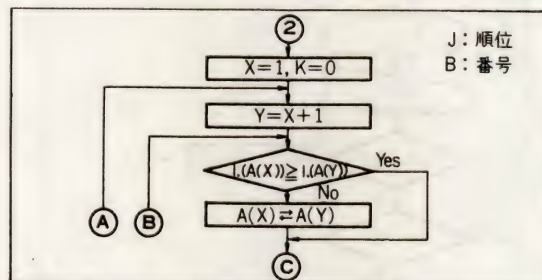
A(2)=80.02

A(3)=70.04

A(4)=100.05

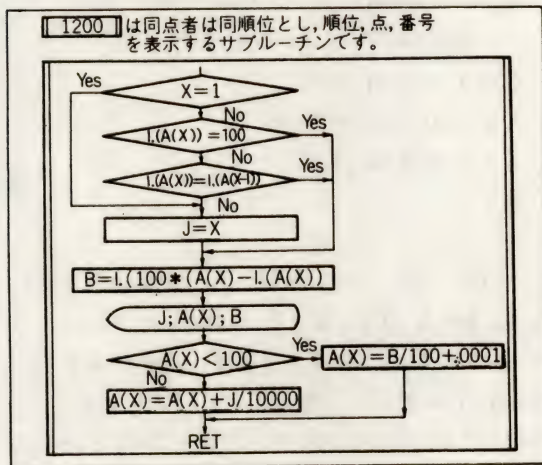
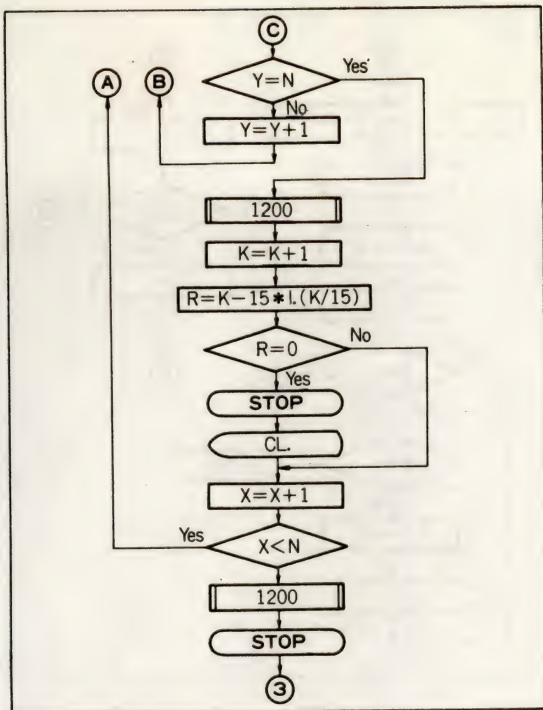
となってメモリに格納されます。

なお，入力ミスの訂正は，誤って999(欠席)を入れたとき，および－100を入れたときは考えていません。

在籍数L，実受験者数N，平均点M，標準偏差S，度数分布ヒストグラム

```
 200 CLEAR
 205 LET Z=T/N: GOSUB 1000
 210 LET M=A
 215 LET Z=SQR(Y/N-M*M): GOSUB 1000
 220 LET S=A
 225 PRINT "ｻﾞｲｾｷ";L;" ｼﾞｭｹﾝｼｬ";N
 230 PRINT "   M=";M;" S=";S
 250 LET A=4096*7+256*14+16*6+1
B=16*3
 255 FOR D=0 TO 10
 260 LET X=A+32*D
 265 IF D=10 THEN  CURSOR 4,14: PICTURE 31,30,30,8F: GOTO 285
 270 LET Y=X+1,Z=X+2,U=X+3,V=X+4,T=X+5,W=B+D
 275 POKE X,W: POKE Y,B: POKE Z,70H
 280 POKE U,W: POKE V,39H: POKE T,8FH
 285 IF B(D)=0 THEN  GOTO 315
 290 FOR I=1 TO B(D)
 295 LET Y=X+5+I
 300 IF Y>X+25 THEN  LET Y=Y-20: POKE Y,80H: GOTO 310
 305 POKE Y,91H
 310 NEXT I
 315 LET Y=4+D: CURSOR 28,Y: PRINT B(D)
 320 NEXT D
 325 STOP

 1000 LET H=100*Z-INT(10*Z)*10
 1005 IF H>=5 THEN  LET A=(INT(10*Z)+1)/10: GOTO 1015
 1010 LET A=INT(10*Z)/10
 1015 RETURN
```

並びかえは，A(1)の整数部(つまり点数)をA(2)，A(3)…と比較していき，自分より大きいものがあれば，その内容を交換するという操作をくり返して，他のすべてのA(Y)と比較が終わればA(1)は最高点のものとしてサブルーチン1200に送られるわけです。

A(1)がすむと，今度はA(2)について，A(3)以下と同様な比較交換が行われ，このようにして，点数順にサブルーチンに送られ表示されます。

流れ図の終わりの方にもう一つのサブルーチン

1200 は同点者は同順位とし，順位，点，番号を表示するサブルーチンです。

があります。最後のA(N)はもう比較する相手がいませんからそのままサブルーチンへ送るのです。

さて，サブルーチンの説明をしましょう。

同点者は同順位とし，たとえば，100点が2人いたら，順位1が2人続き，その次の人は順位3にしたいわけです。

そのために，最初にサブルーチンに送られてくる，A(1)は当然順位1としてA(2)以下のA(X)は，その前に送られて来たA(X−1)と整数部が同じかどうかの比較をして，同じ場合は，J(順番)を変えずに表示部へ移しています。なお，100点のものについて，わざわざ判定をしているのは，100点の人は表示が終わったら，整数部を0にして貯蔵していますから，これがないと同じ100点でも順位が2，3…となってしまいます。

さて，B(番号)ですが，たとえば，A(4)=70.01と小数部が番号ですから，100＊ (A(X)−I.(A(X)))でよいだけですが，マイコンの丸め計算のために，これでは小数部が正しく取り出されませんので注意してください。

表示が終わってから，次の番号順への並べかえのために，順位，点数，番号をまとめて貯蔵しておくのですが，サブルーチンへ送られて来たA(X)の内容の小数第3，4位に順位を入れておきます。

(1)でA(1)=70.01
　　A(2)=80.02
　　A(3)=70.04
　　A(4)=100.05

であったものが，この(2)を通過した段階では，

　　A(1)=0.0501
　　A(2)=80.0202

```
  400 CLEAR
  405 LET X=1,K=0
  410 LET Y=X+1
  415 IF INT(A(X))>=INT(A(Y)) TH
EN  GOTO 425
  420 LET Z=A(X),A(X)=A(Y),A(Y)=
Z
  425 IF Y<N THEN  LET Y=Y+1: GO
TO 415
  430 GOSUB 1200
  435 LET K=K+1
  440 LET R=K-15*INT(K/15)
  445 IF R<>0 THEN  GOTO 460
  450 STOP
  455 CLEAR
  460 LET X=X+1
  475 IF X<N THEN  GOTO 410
  480 GOSUB 1200
  485 STOP
```

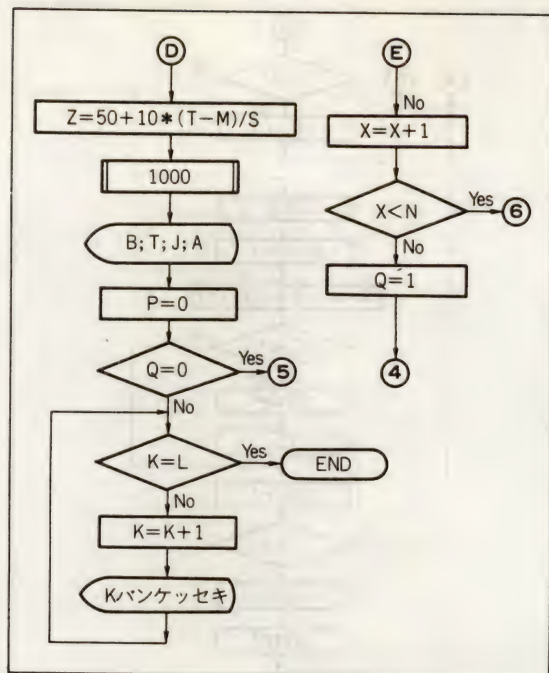
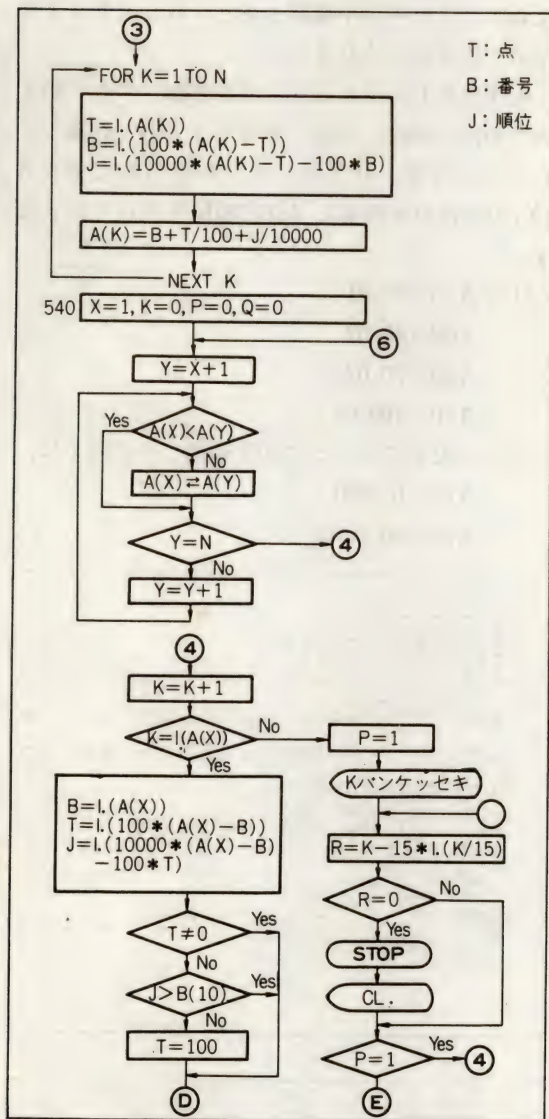
```
 1200 IF X=1 THEN  GOTO 1215
 1205 IF INT(A(X))=100 THEN  GOT
O 1220
 1210 IF INT(A(X))=INT(A(X-1))·T
HEN  GOTO 1220
 1215 LET J=X
 1220 LET B=100*(A(X)-INT(A(X)))

 1225 IF B-INT(B)>.5 THEN  LET B
=INT(B)+1: GOTO 1240
 1230 LET B=INT(B)
 1240 PRINT J;TAB(4);"ｲ";INT(A(X
));TAB(10);"ﾃﾝ --";B;TAB(19);"ﾊﾞ
ﾝ"
 1245 IF A(X)<100 THEN  LET A(X)
=A(X)+J/10000: GOTO 1255
 1250 LET A(X)=B/100+.0001
 1255 RETURN
>
>
> _
```

A(3)=70.0103

A(4)=70.0403

となっています。

番号順に並べかえ，番号 点数，順位，偏差値を表示

まず，(2)で作られたA(X)の整数部(点数)と，小数第1，2位（番号）を入れかえ，それから2と同じように大小の比較をし，今度は小さい順(番号1，2，3……の順）に並びかえるわけです。

(2)のA（X）	A(X)の作りかえ
A(1)=0.0501	A(1)=5.0001
A(2)=80.0202	A(2)=2.8002
A(3)=70.0103	A(3)=1.7003
A(4)=70.0403	A(4)=4.7003

並べかえ

K＝1 A(1)=1.7003

K＝2 A(2)=2.8002

K＝3 A(3)=4.7003

K＝4

K＝5 A(4)=5.0001

次は番号探しです。並べかえの結果小さい方から順にK=1，A(1)；K=2，A(2)……とフローチャート④以下に送られていきます。欠番があると，KとA(X)の整理部（番号）がくい違ってきます。そのときは，そのKが欠席者の番号です。それを表示した後Kだけ1増して再びA(X)の整数部と比較させるわけです。

次に，点数部が0のものは，順位が100点の度数B(10)以内であるかどうかで，100点なのか，本当の0点なのかを判別させます。

実受験者の最後の人は，Q=1とした上で④へ入ります。その番号以降に欠席者がある場合，たとえば上の例では，6番が在籍の最後，つまりL=6で，その番号は欠席という場合もありますので，欠番探しを最後のところにもいれてあります。

```
  500 CLEAR
  505 FOR K=1 TO N
  510 LET T=INT(A(K)),B=INT(100*
(A(K)-T))
  515 LET J=10000*(A(K)-T)-100*B

  520 IF J-INT(J)>.5 THEN  LET J
=INT(J)+1: GOTO 530
  525 LET J=INT(J)
  530 LET A(K)=B+T/100+J/10000
  535 NEXT K
  540 LET X=1,K=0,P=0,Q=0
  545 LET Y=X+1
  550 IF A(X)<A(Y) THEN  GOTO 56
0
  555 LET Z=A(X),A(X)=A(Y),A(Y)=
Z
  560 IF Y=N THEN  GOTO 600
  565 LET Y=Y+1: GOTO 550
  600 LET K=K+1
  605 IF K<>INT(A(X)) THEN  GOTO
 700
  615 LET B=INT(A(X))
  620 LET T=INT(100*(A(X)-B))
  625 LET J=10000*(A(X)-B)-100*T

  630 IF J-INT(J)>.5 THEN  LET J
=INT(J)+1: GOTO 640
  635 LET J=INT(J)
  640 IF T<>0 THEN  GOTO 655
  645 IF J>B(10) THEN  GOTO 655
  650 LET T=100
  655 LET Z=50+10*(T-M)/S
  660 GOSUB 1000
  665 PRINT " ": PRINT B;TAB(4);
"ハ゛ン";T;TAB(12);"テン";J;TAB(18);"
イ  S=";A;
  670 LET P=0
  675 IF Q=0 THEN  GOTO 710
  680 LET K=K+1
  690 PRINT " ": PRINT K;TAB(4);
"ハ゛ン ケッセキ";: GOTO 680
  700 LET P=1: PRINT " ": PRINT
K;TAB(4);"ハ゛ン ケッセキ";
  710 LET R=K-15*INT(K/15)
  715 IF R<>0 THEN  GOTO 730
  720 STOP
  725 CLEAR
  730 IF P=1 THEN  GOTO 600
  735 LET X=X+1
  740 IF X<N THEN  GOTO 545
  745 LET Q=1: GOTO 600
```

30

画面利用のアイデア

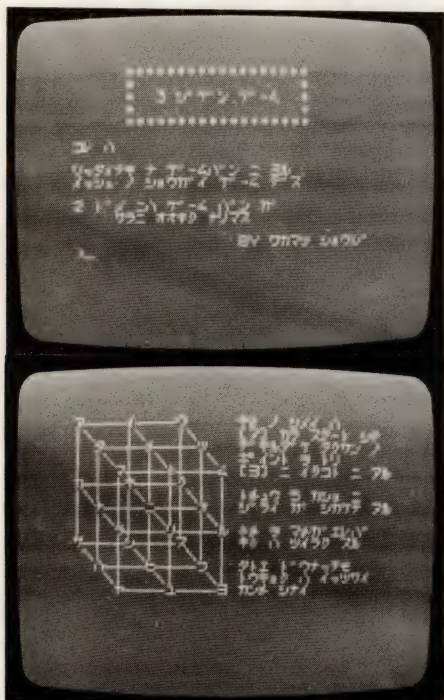
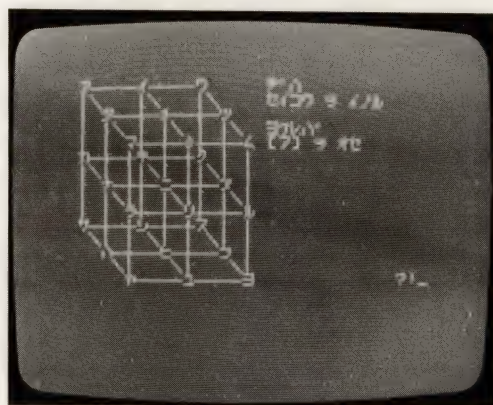
三次元ゲーム・プログラム

TK-80BS＋M20K

若松正司

平面的な作図や，ゲームに飽き足りず，と言って，PETの様な三次元グラフィックの画けない口惜しさから，何とか知恵を絞って，こんなものを作って見ました。これでも，一応三次元になっている処がミソです。

遊び方

タイトルが終ると，左側に立体的な，ジャングルジムの様なゲーム盤，右側に説明文が出ます。

頃合いを計って

となります。

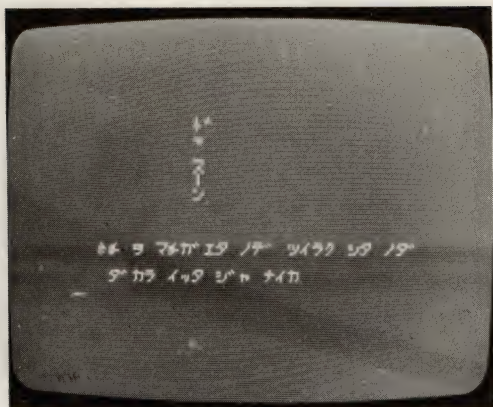
アを押すと，いよいよゲーム開始です。ア以外のキイは，それを受付けません。また，以後，全て操作はカナ文字だけで行いますので，カナは押し放しでよいわけです。

［ア］の部分に四角のポイントが画かれこれでポイントが一つ,獲得されたことになります。地雷は［ア］と，終着点には，仕掛けないので，どんな不運な場合でも，最低1ポイントだけはとれるわけです。

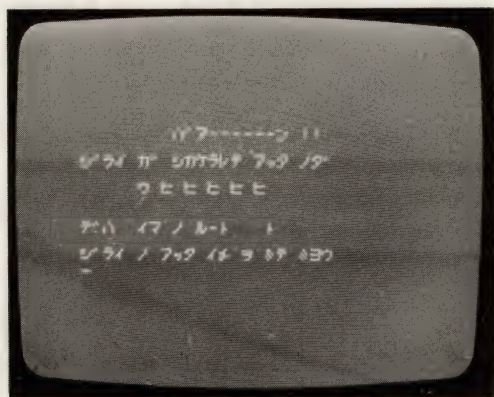
次に，ツギ　ハ？　と聞いて来ます。

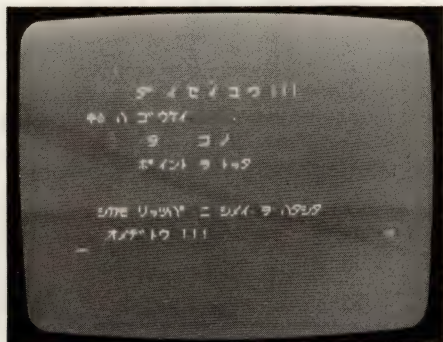
以下，進みたい先のカナ文字をインプットすれば，ポイントが進むのですが，よく画面を見て，立方体であることを正しく観察しないと，道を踏み外して，墜落の憂き目に会います。

その場合は，復改を押した途端に，画面がスク

ロールされ，落下の気分を味わせられ，

などのコメントが現われます。

また，いくら正しい道を選んでも，説明文にあったように，途中5か所に乱数で仕掛けた地雷が待受けていますから，余り沢山ポイントを取ろうと欲張ると，失敗します。

ただし，地雷のあるポイントを選びますと，復改と同時に，画面がクリアされ白黒が一時反転し中心部から亀裂が生じ，四方に拡がりそして消えます。

というコメントが現われます。

しかし，無事，目的地に着いた場合は

という祝福の言葉を添えて，獲得したポイント数が表示されます。

以上，いずれの場合も，あとにおまけがついていて，

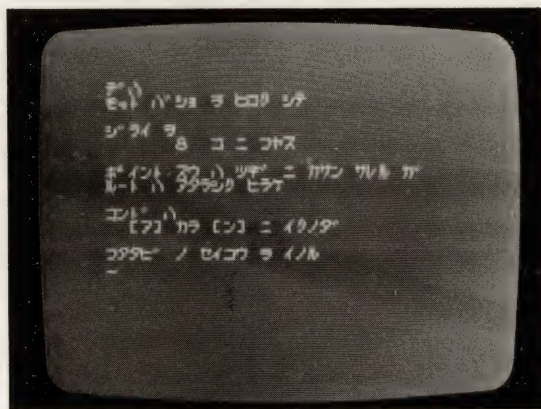
このあと，再び　画面が現われ，今通って来た道が，先ず再現されます。そして，地雷の設置されていたポイントは，⧗の形になって，ゆらゆらと揺れ動き，ここが，地雷だと教えてくれます。失敗した場合は，ああ，ここを通らなければよかったのにとか，成功した場合も，ああ，ここで左を選んだら，アウトだったな，と胸を撫で下したりするわけです。

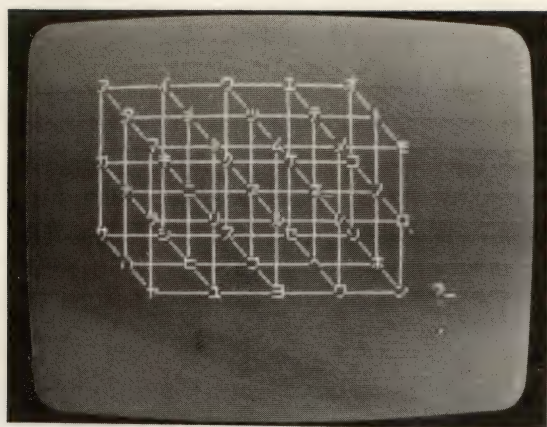
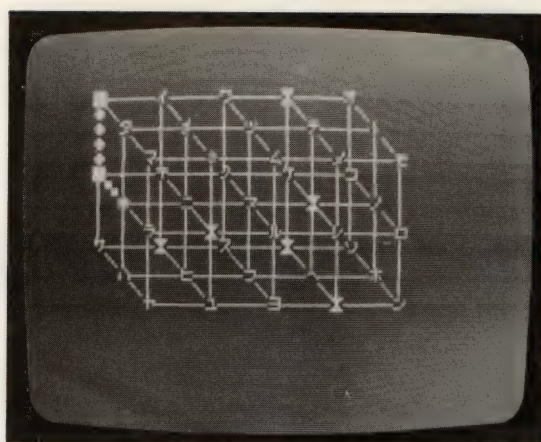
おまけは，まだあって，道を間違えて墜落した時は，その間違いのポイントが，妙な形で動き，あざ笑います。

また，爆発した場合は，その最後のポイントがまるで爆発を思わせるような明滅をしばらく繰返します。

このあと，

モット　ツヅケタイカ？　と聞いてきます。ハイ，と答えれば，当然，もとにもどるわけですが，失敗したあとでも最初の説明文はもう出ません。［ア］をインプットする所から始められ，ポイント数はクリアされています。

また成功している場右は，

というコメントに続き，**写真30―1**のような拡張されたゲーム盤が現われます。

まず，［ア］をインプットすると，あとは全く，今までの要領で進みます。無事［ン］に到達できれば，ポイント数は加算され，大きい方のゲーム盤に戻って来ます。

《写真30-1》　拡張されたゲーム盤

しかし地雷に会わずに三回以上続ける事は至難のようです。

プログラムについて

何バイト要るのか，正確には分かりませんが，SAVEHコマンドで調べて見ると，プログラムだけで，8002からAA69まで使っています。M20Kでのメモリの拡張は不可決のようです。

その代り，のびのびとプログラムしました。REMを省略したり，IF文以外をもっとマルチステートメントにすれば，多少は短くなるのでしょうが余り意味がないと思います。

10～90 タイトルです。

700―900 説明文表示

準備作業が多いせいか，RUNのあと，画面が現われるまで，少し時間がかかるのでその間を利用しました。

100～190

DIM文と初期値の説明（**第30-1表**）。

200―290 READ文で，例えば[ア]は，その値が1，[ン]は225です。これにより正しいルートかどうかを計算できることになります。

300―488 ゲーム盤表示ルーチンです。

IF　S=1……という行番号が，一つ置きにありますが，二回目以後から，大きなゲーム盤にするためのものです。

このゲーム盤を画いた時，このゲームの着想が浮かびました。

A(45)	ポイントが〔ア〕から〔ン〕までありますか，その値。
A$(45)	ポイントの名称〔ア〕～〔ン〕
B(8)	地雷の確保
C(30)，D(30)	斜めに進んだ時の印の位置記憶
E(144)，F(144)	横に進んだ時の印の位置記憶
G(120)，H(120)	縦に進んだ時の印の位置記憶
P(8)，Q(8)	地雷の位置の記憶円
L(38)，M(38)	獲得したポイントの位置記憶用

S　成功すれば1になる	O Q R　経路を埋める印のNomboring
M　現在のポイントの値	V　縦経路の表示が難かしいので補助
P　獲得ポイント数	Z＝失敗した時は10になる
K　何番目にとったポイントか	

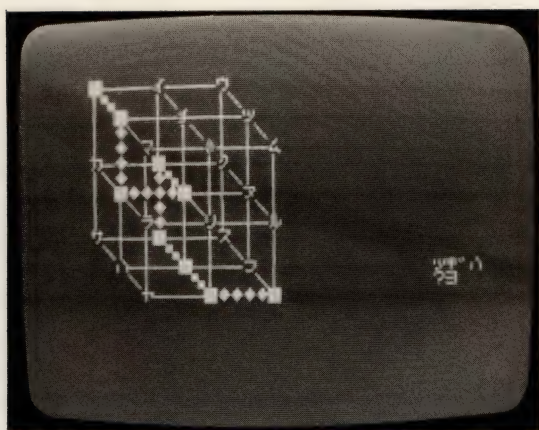
《第30-1表》DIM文と初期値

500―650 地雷設置です。

ポイントの値が3桁でできているので100の桁のAと，10の桁のBと，1の桁のCを乱数で決めて，進んで来たポイントの値と，照合できるようにして置きます。

5個設置するつもりでも，たまたまランダム計算が重なると，4個以下の時もあります。ゲーム盤が大きくなるとS＊3つまり3個だけ増えて，8個になります。

《写真30-2》最初の三次元

《写真30-3》爆発

700—900 ［ア］ のインプット

1000～1050 インプットした，文字と配列の中の文字と，一致した所で次に移ります。

1060～1260 そのポイントは，今までいたポイントから，行ける所かどうかをチェックします。

ただし，正しくなかったら，墜落ルーチンへ飛び，正しければ，地雷があるかどうかを調べるルーチンに飛びます。そして，それも無事ならば，行番号1200で，その位置に進んだ印を画きます。

(1210と1220はゴールなら成功表示ルーチンへ飛ぶことができる)。

1250で次のインプットをきいて，ループします。

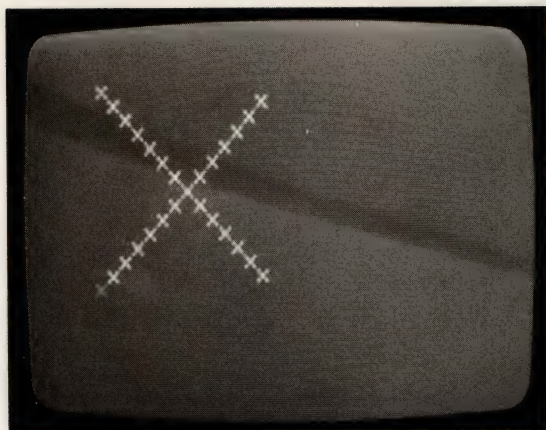
1300—1430 地雷の位置と照合して同じならば爆発ルーチンに飛びます。無事ならば，1070～1090に戻って，正常な進行を続けます。

1500～1790 ポイントを進める時，ポイント間の経路を画くルーチンで，更に後で再現する時にも使うため，記憶させる所でもあります。横も，斜めも，縦も，同じ条件で画くはずなのですが，縦がどうしてもうまく再現できず苦肉の策として1785の処理をしました。

2000～2190 墜落ルーチンです。

2500～2680 これまでにも度々，ここへ飛んで来ましたが，ポイントの値を計算し，それにより，CURSORの位置を決めるルーチンです。

3000～3300 爆発ルーチンです。

一瞬１白黒反転し，中心から，×印で亀裂が散ります。

このPOKE文はマニュアルによると，7DFFHに00Hや02Hを入れればよさそうに思い，7DFEHであるということがわかるまで悩みました。この点，もう少しマニュアルが親切だといいと思います。

途中でルートと地雷の再現へ飛びますが戻ってから，3180以後は爆発点を明滅させる所です。種々なキャラクターを次々と続けて，この感じを出しています。

4000～5111 再現ルーチンです。

このうち5000からは，地雷位置を揺れ動かせて見せる部分です。

6000～6210 継続するかどうかを訊いて，続ける場合に初期値をクリアすべきはしてループします。6110からは，2回目に入った時の説明文です。

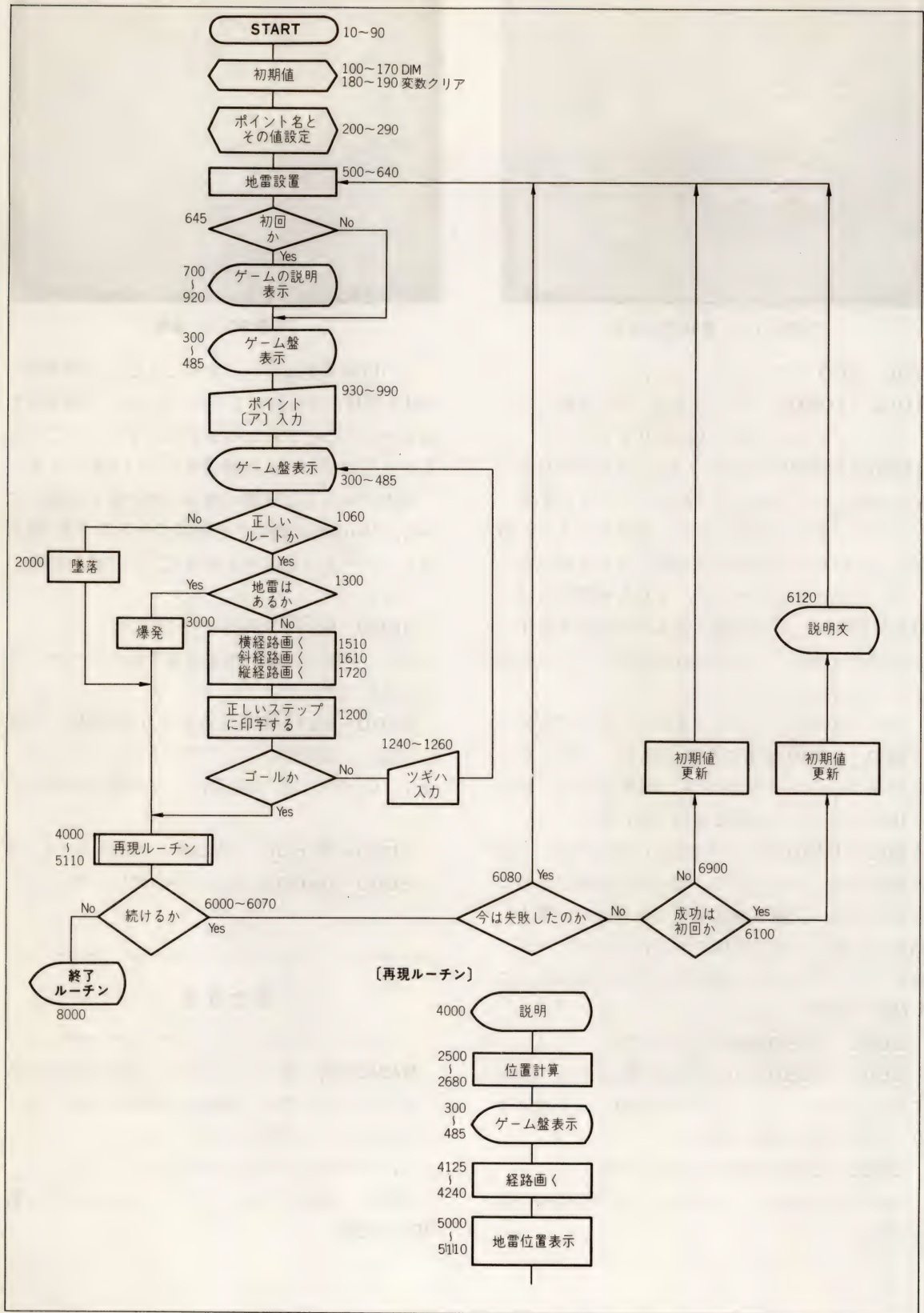
7000～7160 ゴールに着いた時のコメント。

8000～8400 続けない時のオワリのコメントです。

あとがき

BASICで書きましたが，殆ど，時間のロスは感じない位です。然し，機械語を併用すれば，もっと能率がよくなる部分もあるでしょう。

又，効果音を出すのも面白いでしょう。

最後に，簡単なフローチャートを添えます（第30—１図）。

《写真30-1》三次元の表示

```
  10 CLEAR
  30 PRINT "      ********************"
  40 PRINT "      *                  *"
  50 PRINT "      *  3 ｼﾞｹﾞﾝ ｹﾞｰﾑ  *"
  60 PRINT "      *                  *"
  70 PRINT "      ********************"
  80 PRINT "": PRINT "ｺﾚ ﾊ": PRINT
  82 PRINT "ﾘｯﾀｲﾃｷ ﾅ ｹﾞｰﾑﾊﾞﾝ ﾆ ﾖﾙ"
  84 PRINT "ｲｯｼｭ ﾉ ｼｮｳｶﾞｲ ｹﾞｰﾑ ﾃﾞｽ"
  85 PRINT
  86 PRINT "2 ﾄﾞﾒ ﾆﾊ ｹﾞｰﾑ ﾊﾞﾝ ｶﾞ"
  88 PRINT "     ｻﾗﾆ ｵｵｷｸ ﾅﾘﾏｽ"
  90 PRINT "": PRINT TAB(16);"BY ﾜｶﾏﾂ ｼｮｳｼﾞ"
 100 REM
 110 REM ｼｮｷﾁ
 120 REM
 130 DIM A(45),B(8),C(30),D(30)
 140 DIM E(144),F(144),G(120),H(120)
 150 DIM L(38),M(38),P(8),Q(8)
 170 DIM A$(45)
 180 LET S=0,M=0,P=0,K=0,O=0,Q=0,R=0
 190 LET V=0,Z=0
 200 REM
 210 REM ﾘｰﾄﾞ
 220 REM
 230 FOR I=1 TO 45: READ A(I): NEXT I
 240 DATA 1,2,3,4,5,101,102,103,104,105,201,202,203,204,205
 245 DATA 11,12,13,14,15,111,112,113,114,115,211,212,213,214,215
 250 DATA 21,22,23,24,25,121,122,123,124,125,221,222,223,224,225
 260 FOR I=1 TO 45: READ A$(I): NEXT I
 270 DATA ｱ,ｲ,ｳ,ｴ,ｵ,ｶ,ｷ,ｸ,ｹ,ｺ,ｻ,ｼ,ｽ,ｾ,ｿ
 275 DATA ﾀ,ﾁ,ﾂ,ﾃ,ﾄ,ﾅ,ﾆ,ﾇ,ﾈ,ﾉ,ﾊ,ﾋ,ﾌ,ﾍ,ﾎ
 280 DATA ﾏ,ﾐ,ﾑ,ﾒ,ﾓ,ﾗ,ﾘ,ﾙ,ﾚ,ﾛ,ﾔ,ﾕ,ﾖ,ﾜ,ﾝ
 290 GOTO 500
 300 REM
 310 REM ｹﾞｰﾑﾊﾞﾝ ﾋｮｳｼﾞ
 320 REM
 330 CURSOR 1,1: PICTURE 71,9A,9A,9A,9A,72,9A,9A,9A,9A,73
 340 IF S=1 THEN  CURSOR 12,1: PICTURE 9A,9A,9A,9A,74,9A,9A,9A,9A,75
 350 CURSOR 1,2: PICTURE 8B,B6,20,20,20,8B,B6,20,20,20,8B,B6
 355 IF S=1 THEN  CURSOR 13,2: PICTURE 20,20,20,8B,B6,20,20,20,8B,B6
 360 CURSOR 1,3: PICTURE 8B,20,40,9A,9A,A4,9A,41,9A,9A,A4,9A,42
 365 IF S=1 THEN  CURSOR 14,3: PICTURE 9A,9A,A4,9A,43,9A,9A,A4,9A,44
 370 CURSOR 1,4: PICTURE 8B,20,8B,B6,20,8B,20,8B,B6,20,8B,20,8B,B6
 375 IF S=1 THEN  CURSOR 15,4: PICTURE 20,8B,20,8B,B6,20,8B,20,8B,B6
 380 CURSOR 1,5: PICTURE 8B,20,8B,20,4F,A4,9A,A4,9A,50,A4,9A,A4,9A,51
 385 IF S=1 THEN  CURSOR 16,5: PICTURE A4,9A,A4,9A,52,A4,9A,A4,9A,53
 390 CURSOR 1,6: PICTURE 76,9A,A4,9A,A4,77,9A,A4,9A,A4,78,20,8B,20,8B
 395 IF S=1 THEN  CURSOR 12,6: PICTURE 9A,A4,9A,A4,79,9A,A4,9A,A4,7A,20,8B,20,8
B
 400 CURSOR 1,7: PICTURE 8B,B6,8B,20,8B,8B,B6,8B,20,8B,8B,B6,8B,20,8B
 405 IF S=1 THEN  CURSOR 16,7: PICTURE 8B,B6,8B,20,8B,8B,B6,8B,20,8B
 410 CURSOR 1,8: PICTURE 8B,20,45,9A,A4,A4,9A,46,9A,A4,A4,9A,47,20,8B
 415 IF S=1 THEN  CURSOR 14,8: PICTURE 9A,A4,A4,9A,48,9A,A4,A4,9A,49,20,8B
 420 CURSOR 1,9: PICTURE 8B,20,8B,B6,8B,8B,20,8B,B6,8B,8B,20,8B,B6,8B
 425 IF S=1 THEN  CURSOR 16,9: PICTURE 8B,20,8B,B6,8B,8B,20,8B,B6,8B
 430 CURSOR 1,10: PICTURE 8B,20,8B,20,57,A4,9A,A4,9A,58,A4,9A,A4,9A,59
 435 IF S=1 THEN  CURSOR 16,10: PICTURE A4,9A,A4,9A,5A,A4,9A,A4,9A,5B
 440 CURSOR 1,11: PICTURE 7B,9A,A4,9A,A4,7C,9A,A4,9A,A4,7D,20,8B,20,8B
 445 IF S=1 THEN  CURSOR 12,11: PICTURE 9A,A4,9A,A4,7E,9A,A4,9A,A4,7F,20,8B,20,
8B
 450 CURSOR 2,12: PICTURE B6,8B,20,8B,20,B6,8B,20,8B,20,B6,8B,20,8B
 455 IF S=1 THEN  CURSOR 16,12: PICTURE 20,B6,8B,20,8B,20,B6,8B,20,8B
 460 CURSOR 3,13: PICTURE 4A,9A,A4,9A,9A,4B,9A,A4,9A,9A,4C,20,8B
 465 IF S=1 THEN  CURSOR 14,13: PICTURE 9A,A4,9A,9A,4D,9A,A4,9A,9A,4E,20,8B
 470 CURSOR 4,14: PICTURE B6,8B,20,20,20,B6,8B,20,20,20,B6,8B
 475 IF S=1 THEN  CURSOR 16,14: PICTURE 20,20,20,B6,8B,20,20,20,B6,8B
 480 CURSOR 5,15: PICTURE 54,9A,9A,9A,9A,55,9A,9A,9A,9A,56
 485 IF S=1 THEN  CURSOR 16,15: PICTURE 9A,9A,9A,9A,5C,9A,9A,9A,9A,5D
 488 RETURN
 500 REM
 510 REM ｼﾞﾗｲ ｼｶｹ
 520 REM
 530 FOR I=1 TO 5+S*3
 540 RANDOMIZE
 550 LET A=INT(RND(2)+.5)
 560 LET B=INT(RND(2)+.5)
 570 LET C=INT(RND(2+S*2)+.5)+1
 580 LET B(I)=A*100+B*10+C
 590 IF B(I)=1 THEN 540
 600 IF B(I)=225 THEN 540
 610 IF B(I)=223 THEN  IF S=0 THEN 540
 620 LET P(I)=1+(C-1)*5+B*2
 630 LET Q(I)=1+A*5+B*2
 640 NEXT I
 645 IF Z=10 THEN  CLEAR : GOTO 930
 650 IF S>=1 THEN  CLEAR : GOTO 950
```

```
700 REM
710 REM ｹﾞｰﾑ ﾉ ｾﾂﾒｲ
720 REM
730 CLEAR
740 PRINT TAB(16);"ｷﾐ ﾉ ｼﾒｲ ﾊ"
750 PRINT TAB(16);"[ｱ] ｶﾗ ｽﾀｰﾄ ｼﾃ"
760 PRINT TAB(16);"ﾃﾞｷﾙﾀﾞｹ ﾀｸｻﾝ ﾉ"
770 PRINT TAB(16);"ﾎﾟｲﾝﾄ ｦ ﾄﾘ"
780 PRINT TAB(16);"[ﾖ] ﾆ ｲｸｺﾄ ﾆ ｱﾙ"
790 PRINT
800 PRINT TAB(16);"ﾄﾁｭｳ 5 ｶｼｮ ﾆ"
810 PRINT TAB(16);"ｼﾞﾗｲ ｶﾞ ｼｶｹﾃ ｱﾙ"
820 PRINT
830 PRINT TAB(16);"ﾐﾁ ｦ ﾏﾁｶﾞｴﾚﾊﾞ"
840 PRINT TAB(16);"ｷﾐ ﾊ ﾂｲﾗｸ ｽﾙ"
850 PRINT
860 PRINT TAB(16);"ﾀﾄｴ ﾄﾞｳﾅｯﾃﾓ "
870 PRINT TAB(16);"ﾄｳｷｮｸ ﾊ ｲｯｻｲ"
880 PRINT TAB(16);"ｶﾝﾁ ｼﾅｲ"
890 GOSUB 300
900 FOR T=1 TO 1200: NEXT T: CLEAR
910 PRINT TAB(16);"ﾃﾞﾊ"
920 PRINT TAB(16);"ｾｲｺｳ ｦ ｲﾉﾙ": PRINT
930 PRINT TAB(16);"ﾖｹﾚﾊﾞ"
935 LET Z=0
940 PRINT TAB(16);"[ｱ] ｦ ｵｾ"
950 GOSUB 300
960 CURSOR 28,15: INPUT ""X$
970 IF X$<>"ｱ" THEN 960
980 CLEAR
990 GOSUB 300
1000 REM
1010 REM ｽﾃｯﾌﾟ ｦ ｵｺﾅｳ
1020 REM
1030 FOR I=1 TO 45
1040 IF X$=A$(I) THEN  LET N=A(I): GOTO 1060
1050 NEXT I
1060 REM ﾀﾀﾞｼｲ ﾙｰﾄ ｶ
1070 IF ABS(N-M)=1 THEN  GOSUB 1300: GOSUB 1500: GOTO 1200
1080 IF ABS(N-M)=10 THEN  GOSUB 1300: GOSUB 1600: GOTO 1200
1090 IF ABS(N-M)=100 THEN  GOSUB 1300: GOSUB 1700: GOTO 1200
1100 GOTO 2000
1200 CURSOR X,Y: PICTURE 80
1205 LET M=N,P=P+1
1210 IF N=225 THEN 7000
1220 IF N=223 THEN  IF S=0 THEN 7000
1240 CURSOR 28,13: PICTURE 42,77,5E,4A
1250 CURSOR 28,14: INPUT ""X$
1260 GOTO 1000
1300 REM
1310 REM ｼﾞﾗｲ ﾊ ｱﾙｶ
1320 REM
1330 FOR I=1 TO 5+S*3
1340 IF N=B(I) THEN 3000
1350 NEXT I
1360 GOSUB 2500
1410 LET K=K+1
1420 LET L(K)=X,M(K)=Y
1430 RETURN
1500 REM
1510 REM ﾖｺ ｹｲﾛ
1515 IF N=1 THEN  RETURN
1520 FOR I=4 TO 1 STEP -1
1530 LET A1=X-(I*(N-M))
1540 CURSOR A1,Y: PICTURE D4
1550 LET O=O+1
1560 LET E(O)=A1,F(O)=Y
1570 NEXT I
1580 RETURN
1600 REM
1610 REM ﾅﾅﾒ ｹｲﾛ
1620 LET A2=X-1*((N-M)/10)
1630 LET B2=Y-1*((N-M)/10)
1640 CURSOR A2,B2: PICTURE A2
1650 LET Q=Q+1
1660 LET C(Q)=A2,D(Q)=B2
1670 RETURN
```

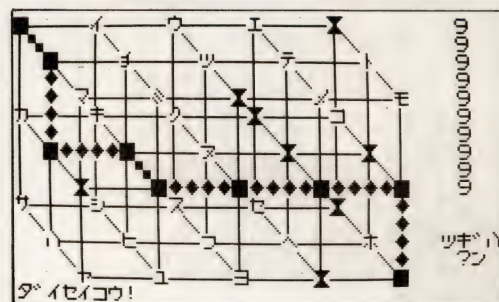
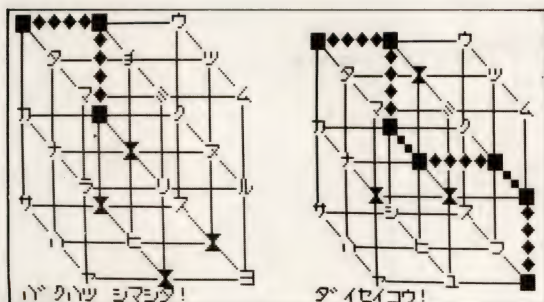
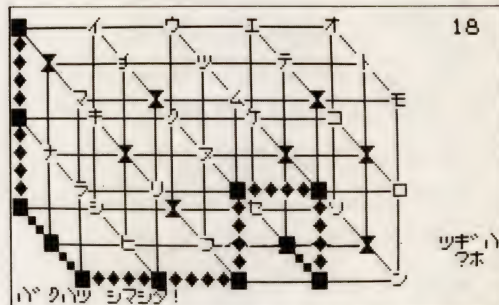
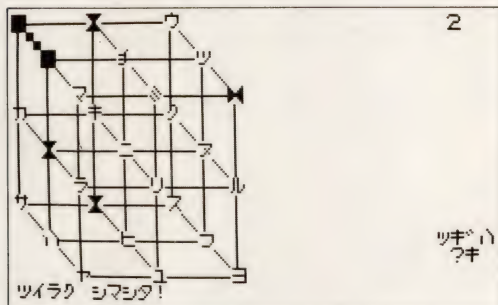
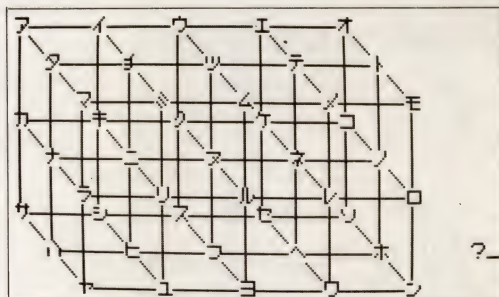
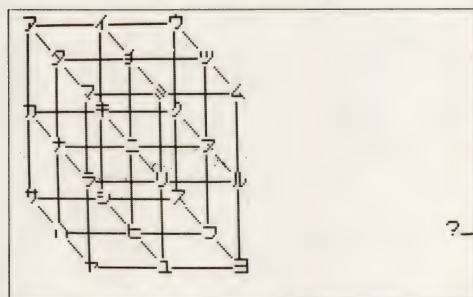
```
1700 REM
1720 REM ﾀﾃ ｹｲﾛ
1730 FOR I=4 TO 1 STEP -1
1740 LET B3=Y-I*((N-M)/100)
1750 CURSOR X,B3: PICTURE D4
1760 LET R=R+1
1770 LET G(R)=X,H(R)=B3
1780 NEXT I
1785 LET V=V+1
1790 RETURN
2000 REM
2010 REM ﾂｲﾗｸ ﾙｰﾁﾝ
2020 REM
2025 LET Z=10
2030 FOR I=1 TO 16
2040 CURSOR 1,16: PRINT
2050 NEXT I
2060 PRINT TAB(10);"ﾄﾞ"
2070 PRINT TAB(10);"ｯ": PRINT
2080 PRINT TAB(10);"ｽ"
2090 PRINT TAB(10);"I"
2100 PRINT TAB(10);"ﾝ"
2110 PRINT "": PRINT "": PRINT
2120 PRINT "  ﾐﾁ ｦ ﾏﾁｶﾞｴﾀ ﾉﾃﾞ ﾂｲﾗｸ ｼﾀ ﾉﾀﾞ"
2130 PRINT "": PRINT "   ﾀﾞｶﾗ ｲｯﾀ ｼﾞｬ ﾅｲｶ"
2140 FOR T=1 TO 600: NEXT T
2150 GOSUB 4000
2155 GOSUB 2500
2160 FOR S=1 TO 30
2163 CURSOR X,Y: PICTURE FF
2165 FOR T=1 TO 20: NEXT T
2168 CURSOR X,Y: PICTURE B9
2170 FOR T=1 TO 20: NEXT T
2180 NEXT S
2190 GOTO 6000
2500 REM
2510 REM N ﾉ ｲﾁ ｹｲｻﾝ
2520 REM
2530 LET A=INT(N/100)
2540 LET B=INT((N-A*100)/10)
2550 LET C=N-A*100-B*10
2560 LET X=1+(C-1)*5+B*2
2570 LET Y=1+A*5+B*2
2680 RETURN
3000 REM
3010 REM ﾊﾞｸﾊﾂ ﾙｰﾁﾝ
3020 REM
3030 LET Z=10
3040 POKE 7DFEH,00H
3050 CLEAR
3060 LET X=8+8*S,Y=8
3070 CURSOR X,Y: PICTURE 18
3075 FOR R1=1 TO 7
3080 FOR P1=X-R1 TO X+R1 STEP R1*2
3085 FOR Q1=Y-R1 TO Y+R1 STEP R1*2
3090 CURSOR P1,Q1: PICTURE 18
3095 NEXT Q1: NEXT P1: NEXT R1
3100 POKE 7DFEH,02H
3105 CLEAR : PRINT "": PRINT "": PRINT "": PRINT
3110 PRINT TAB(8);"ﾊﾞｱｰｰｰｰｰｰﾝ !!": PRINT
3115 PRINT "ｼﾞﾗｲ ｶﾞ ｼｶｹﾗﾚﾃ ｱｯﾀ ﾉﾀﾞ"
3120 PRINT "": PRINT "     ｳ ﾋ ﾋ ﾋ ﾋ ﾋ"
3125 FOR T=0 TO 500: NEXT T
3128 GOSUB 4000
3140 IF ABS(N-M)=1 THEN  GOSUB 1500: GOTO 3180
3150 IF ABS(N-M)=10 THEN  GOSUB 1600: GOTO 3180
3160 GOSUB 1700
3165 GOSUB 2500
3180 FOR B=1 TO 20
3190 CURSOR X,Y: PICTURE CC
3200 FOR T=1 TO 10: NEXT T
3210 CURSOR X,Y: PICTURE DF
3220 FOR T=1 TO 10: NEXT T
3230 CURSOR X,Y: PICTURE CA
3240 FOR T=1 TO 10: NEXT T
3250 CURSOR X,Y: PICTURE DE
3260 FOR T=1 TO 10
3270 CURSOR X,Y: PICTURE CB
3280 FOR T=1 TO 10: NEXT T
3290 NEXT B
3300 GOTO 6000
4000 REM
4010 REM ﾙｰﾄ ﾄ ｼﾞﾗｲ ﾉ ｻｲｹﾞﾝ
4020 REM
4025 PRINT "": PRINT
4030 PRINT "ﾃﾞﾊ  ｲﾏ ﾉ ﾙｰﾄ   ﾄ"
4035 PRINT
4040 PRINT "ｼﾞﾗｲ ﾉ ｱｯﾀ ｲﾁ ｦ ﾐﾃ ﾐﾖｳ"
```

```
4050 FOR T=1 TO 600: NEXT T
4100 CLEAR
4130 FOR I=1 TO K
4140 CURSOR L(I),M(I): PICTURE 80
4150 NEXT I
4155 IF O=0 THEN 4185
4160 FOR I=1 TO O
4170 CURSOR E(I),F(I): PICTURE CA
4180 NEXT I
4185 IF Q=0 THEN 4215
4190 FOR I=1 TO Q
4200 CURSOR C(I),D(I): PICTURE A2
4210 NEXT I
4215 IF R=0 THEN 5000
4220 FOR I=1 TO V*4
4230 CURSOR G(I),H(I): PICTURE CA
4240 NEXT I
5000 REM
5010 REM ｼﾞﾗｲ ｲﾁ ｦ ﾐｾﾙ
5020 REM
5030 FOR B=1 TO 10
5040 FOR J=1 TO 5+S*3
5050 CURSOR P(J),Q(J): PICTURE B7
5060 FOR T=1 TO 5: NEXT T
5070 CURSOR P(J),Q(J): PICTURE B8
5080 FOR T=1 TO 5: NEXT T
5090 NEXT J
5100 NEXT B
5110 RETURN
6000 REM
6010 REM ﾂﾂﾞｹﾙ ｶ
6020 REM
6025 CLEAR
6030 FOR T=1 TO 100: NEXT T
6040 PRINT "ﾓｯﾄ ﾂﾂﾞｹﾀｲ ｶ": PRINT
6050 PRINT TAB(10);"(ﾊｲ / ｲｲｴ)"
6060 PRINT "": PRINT TAB(12);"": INPUT L$
6070 IF L$<>"ﾊｲ" THEN 8000
6080 IF Z=10 THEN  LET M=0,S=0,P=0,K=0,O=0,Q=0,R=0,V=0: GOTO 500
6090 IF S>=1 THEN  LET M=0,O=0,Q=0,R=0,K=0,V=0: GOTO 500
6100 LET S=S+1,M=0,O=0,Q=0,R=0,K=0,V=0
6110 CLEAR
6120 PRINT "ﾃﾞﾊ"
6130 PRINT "ﾓｯﾄ ﾊﾞｼｮ ｦ ﾋﾛｸ ｼﾃ"
6140 PRINT "": PRINT "ｼﾞﾗｲ ｦ"
6150 PRINT TAB(6);"8  ｺ ﾆ ﾌﾔｽ": PRINT
6160 PRINT "ﾎﾟｲﾝﾄ ｽｳ ﾊ ﾂｷﾞ ﾆ ｶｻﾝ ｻﾚﾙ ｶﾞ"
6170 PRINT "ﾙｰﾄ ﾊ ｱﾀﾗｼｸ ﾋﾗｹ"
6180 PRINT "": PRINT "ｺﾝﾄﾞ ﾊ"
6190 PRINT "  [ｱ] ｶﾗ [ﾝ] ﾆ ｲｸﾉﾀﾞ": PRINT
6200 PRINT "ﾌﾀﾀﾋﾞ ﾉ ｾｲｺｳ ｦ ｲﾉﾙ"
6205 FOR T=1 TO 800: NEXT T
6210 GOTO 500
7000 REM
7010 REM ｾｲｺｳ
7020 REM
7030 FOR T=1 TO 400: NEXT T
7035 CLEAR
7040 PRINT "": PRINT "": PRINT "": PRINT "": PRINT
7050 PRINT TAB(6);"ﾀﾞ ｲ ｾ ｲ ｺ ｳ !!!"
7055 PRINT
7060 PRINT " ｷﾐ ﾊ ｺﾞｳｹｲ"
7070 PRINT "": PRINT TAB(6);P;"   ｺ ﾉ"
7080 PRINT "": PRINT TAB(6);"ﾎﾟｲﾝﾄ ｦ ﾄｯﾀ"
7090 PRINT "": PRINT "": PRINT
7100 PRINT "  ｼｶﾓ ﾘｯﾊﾟ ﾆ ｼﾒｲ ｦ ﾊﾀｼﾀ": PRINT
7110 PRINT "    ｵﾒﾃﾞﾄｳ !!!"
7120 FOR T=0 TO 800: NEXT T
7130 GOSUB 4000
7150 FOR T=1 TO 500: NEXT T
7160 GOTO 6000
8000 REM
8010 REM ｵﾜﾘ
8020 REM
8030 PRINT
8060 PRINT "::::::::::::::::::::::::::::"
8070 PRINT "": PRINT "": PRINT "": PRINT
8080 PRINT " ｿﾛｿﾛ ﾍﾞﾝｷｮｳ ｦ ｼﾅｻｲ"
8090 PRINT "": PRINT "ｼｺﾞﾄ ﾓ ｼﾅｸﾁｬ"
8100 PRINT "": PRINT "": PRINT
8200 PRINT " ﾃﾞﾊ      ﾊﾞｲ ﾊﾞｲ･･･"
8300 PRINT "": PRINT "": PRINT
8400 END
```

ミニ三次元ゲーム・プログラム

このプログラムは，前述のオリジナル版を従来のメモリ・スペース（7KB）に納まるように縮少させたものです。機能，たとえば，まず小サイズの次元の無事終了後，直ちに大サイズの次元に挑戦できるといったものは残してありますので，充分に楽しめるでしょう。主な相違異点は，バクハツやツイラクメッセージを同画面上に表示し直ちにそのまま地雷を表示してしまう点，画面右上に各ポイント点数を常に表示している点でしょう。こうする事によって，オリジナル版の配列宣言を半分以下にとどめる事ができましたが，現在でもぎりぎりのスペースですので気をつけて下さい。このプログラムから抜け出るには，各スタート点(ア)の際，ア以外のキーを入れれば良いようになっています。

RUNさせれば分かりますが，地雷の数は常に5，8個ではありません。これは，500行で行なっている乱数に同位置のものが2，3度選ばれてしまうためですが，場合によっては楽しい現象なのでオリジナル版に従いました（5，8個以下になる）。

こういうタイプのゲームは，一発一発じっくりと考えて挑戦していったほうが面白いです。最後まで貫通させて成功するにはかなり骨が折れますよ。音も出せると迫力は増すでしょう。ぜひ試して下さい。

```
>L.
  10 CLEAR
  130 DIM A(45),B(8),P(8),Q(8)
  170 DIM A$(45)
  180 LET S=0,P=0,M=0,Z=0
  200 REM リード
  225 POKE 8624H,0H
  230 FOR I=1 TO 45: READ A(I):
NEXT I
  240 DATA 1,2,3,4,5,101,102,103
,104,105,201,202,203,204,205
  245 DATA 11,12,13,14,15,111,11
2,113,114,115,211,212,213,214,21
5
  250 DATA 21,22,23,24,25,121,12
2,123,124,125,221,222,223,224,22
5
  260 FOR I=1 TO 45: READ A$(I):
 NEXT I
  270 DATA ア,イ,ウ,エ,オ,カ,キ,ク,ケ,コ,サ
,シ,ス,セ,ソ
  275 DATA タ,チ,ツ,テ,ト,ナ,ニ,ヌ,ネ,ノ,ハ
,ヒ,フ,ヘ,ホ
  280 DATA マ,ミ,ム,メ,モ,ラ,リ,ル,レ,ロ,ヤ
,ユ,ヨ,ワ,ン
  290 GOTO 500
  300 REM
  330 CURSOR 1,1: PICTURE 71,9A,
9A,9A,9A,72,9A,9A,9A,9A,73
  340 IF S=1 THEN  CURSOR 12,1:
PICTURE 9A,9A,9A,9A,74,9A,9A,9A,
9A,75
  350 CURSOR 1,2: PICTURE 8B,B6,
20,20,20,8B,B6,20,20,20,8B,B6
  355 IF S=1 THEN  CURSOR 13,2:
PICTURE 20,20,20,8B,B6,20,20,20,
8B,B6
  360 CURSOR 1,3: PICTURE 8B,20,
40,9A,9A,A4,9A,41,9A,9A,A4,9A,42

  365 IF S=1 THEN  CURSOR 14,3:
PICTURE 9A,9A,A4,9A,43,9A,9A,A4,
9A,44
  370 CURSOR 1,4: PICTURE 8B,20,
8B,B6,20,8B,20,8B,B6,20,8B,20,8B
,B6
  375 IF S=1 THEN  CURSOR 15,4:
PICTURE 20,8B,20,8B,B6,20,8B,20,
8B,B6
  380 CURSOR 1,5: PICTURE 8B,20,
8B,20,4F,A4,9A,A4,9A,50,A4,9A,A4
,9A,51
  385 IF S=1 THEN  CURSOR 16,5:
PICTURE A4,9A,A4,9A,52,A4,9A,A4,
9A,53
  390 CURSOR 1,6: PICTURE 76,9A,
A4,9A,A4,77,9A,A4,9A,A4,78,20,8B
,20,8B
  395 IF S=1 THEN  CURSOR 12,6:
PICTURE 9A,A4,9A,A4,79,9A,A4,9A,
A4,7A,20,8B,20,8B
  400 CURSOR 1,7: PICTURE 8B,B6,
8B,20,8B,8B,B6,8B,20,8B,8B,B6,8B
,20,8B
  405 IF S=1 THEN  CURSOR 16,7:
PICTURE 8B,B6,8B,20,8B,8B,B6,8B,
20,8B
  410 CURSOR 1,8: PICTURE 8B,20,
45,9A,A4,A4,9A,46,9A,A4,A4,9A,47
,20,8B
  415 IF S=1 THEN  CURSOR 14,8:
PICTURE 9A,A4,A4,9A,48,9A,A4,A4,
9A,49,20,8B
 420 CURSOR 1,9: PICTURE 8B,20,
8B,B6,8B,8B,20,8B,B6,8B,8B,20,8B
,B6,8B
  425 IF S=1 THEN  CURSOR 16,9:
PICTURE 8B,20,8B,B6,8B,8B,20,8B,
B6,8B
  430 CURSOR 1,10: PICTURE 8B,20
,8B,20,57,A4,9A,A4,9A,58,A4,9A,A
4,9A,59
  435 IF S=1 THEN  CURSOR 16,10:
PICTURE A4,9A,A4,9A,5A,A4,9A,A4
,9A,5B
  440 CURSOR 1,11: PICTURE 7B,9A
,A4,9A,A4,7C,9A,A4,9A,A4,7D,20,8
B,20,8B
  445 IF S=1 THEN  CURSOR 12,11:
 PICTURE 9A,A4,9A,A4,7E,9A,A4,9A
,A4,7F,20,8B,20,8B
  450 CURSOR 2,12: PICTURE B6,8B
,20,8B,20,B6,8B,20,8B,20,B6,8B,2
0,8B
  455 IF S=1 THEN  CURSOR 16,12:
 PICTURE 20,B6,8B,20,8B,20,B6,8B
,20,8B
  460 CURSOR 3,13: PICTURE 4A,9A
,A4,9A,9A,4B,9A,A4,9A,9A,4C,20,8
B
  465 IF S=1 THEN  CURSOR 14,13:
 PICTURE 9A,A4,9A,9A,4D,9A,A4,9A
,9A,4E,20,8B
  470 CURSOR 4,14: PICTURE B6,8B
,20,20,20,B6,8B,20,20,20,B6,8B
  475 IF S=1 THEN  CURSOR 16,14:
 PICTURE 20,20,20,B6,8B,20,20,20
,B6,8B
  480 CURSOR 5,15: PICTURE 54,9A
,9A,9A,9A,55,9A,9A,9A,9A,56
  485 IF S=1 THEN  CURSOR 16,15:
 PICTURE 9A,9A,9A,9A,5C,9A,9A,9A
,9A,5D
  488 RETURN
  500 REM
  530 FOR I=1 TO 5+S*3
  540 RANDOMIZE
  550 LET A=INT(RND(2)+.5)
  560 LET B=INT(RND(2)+.5)
  570 LET C=INT(RND(2+S*2)+.5)+1

  580 LET B(I)=A*100+B*10+C
  590 IF B(I)=1 THEN 540
  600 IF B(I)=225 THEN 540
  610 IF B(I)=223 THEN  IF S=0 T
HEN 540
  620 LET P(I)=1+(C-1)*5+B*3
  630 LET Q(I)=1+A*5+B*2
  640 NEXT I
  950 GOSUB 300
  960 CURSOR 28,13: INPUT " "X$
  970 IF X$<>"ア" THEN  STOP
 1000 REM
 1030 FOR I=1 TO 45
 1040 IF X$=A$(I) THEN  LET N=A(
I): GOTO 1060
 1050 NEXT I
 1060 REM
 1070 IF ABS(N-M)=1 THEN  GOSUB
1300: GOSUB 1500: GOTO 1200
 1080 IF ABS(N-M)=10 THEN  GOSUB
 1300: GOSUB 1600: GOTO 1200
 1090 IF ABS(N-M)=100 THEN  GOSU
B 1300: GOSUB 1700: GOTO 1200
 1100 GOTO 2000
 1200 CURSOR X,Y: PICTURE 80
 1205 LET M=N,P=P+1
 1210 IF N=225 THEN 7000
 1220 IF N=223 THEN  IF S=0 THEN
7000
 1230 CURSOR 28,1: PRINT P;
 1240 CURSOR 28,13: PRINT "ツキ

 1250 CURSOR 28,14: INPUT " "X$
 1255 POKE 8624H,0H
 1260 GOTO 1000
 1300 REM
 1330 FOR I=1 TO 5+S*3
 1340 IF N=B(I) THEN 3000
 1350 NEXT I
 1360 GOSUB 2500
 1430 RETURN
 1500 REM
 1515 IF N=1 THEN  RETURN
 1520 FOR I=4 TO 1 STEP -1
 1530 LET A1=X-(I*(N-M))
 1540 CURSOR A1,Y: PICTURE D4
 1570 NEXT I
 1580 RETURN
 1600 REM
 1620 LET A2=X-1*((N-M)/10)
 1630 LET B2=Y-1*((N-M)/10)
 1640 CURSOR A2,B2: PICTURE A2
 1670 RETURN
 1700 REM
 1730 FOR I=4 TO 1 STEP -1
 1740 LET B3=Y-I*((N-M)/100)
 1750 CURSOR X,B3: PICTURE D4
 1780 NEXT I
 1790 RETURN
 2000 REM
 2025 LET Z=10
 2120 CURSOR 1,16: PRINT "ツイラク シ
マシタ!";
 2190 GOTO 6000
 2500 REM
 2530 LET A=INT(N/100)
 2540 LET B=INT((N-A*100)/10)
 2550 LET C=N-A*100-B*10
 2560 LET X=1+(C-1)*5+B*3
 2570 LET Y=1+A*5+B*2
 2680 RETURN
 3000 REM
 3030 LET Z=10
 3050 CURSOR 1,16: PRINT "ゲキツ
イシマシタ!";
 3300 GOTO 6000
 5000 REM
 5030 FOR B=1 TO 10
 5040 FOR J=1 TO 5+S*3
 5050 CURSOR P(J),Q(J): PICTURE
B7
 5060 FOR T=1 TO 5: NEXT T
 5070 CURSOR P(J),Q(J): PICTURE
B8
 5080 FOR T=1 TO 5: NEXT T
 5090 NEXT J: NEXT B
 5110 RETURN
 6000 REM
 6010 GOSUB 5000
 6015 IF Z=10 THEN  LET S=0,P=0:
GOTO 6080
 6020 LET S=1
 6080 LET M=0,Z=0
 6082 FOR T=1 TO 200: NEXT T
 6085 CLEAR
 6090 GOTO 500
 7000 REM
 7030 FOR W=1 TO 10
 7035 POKE 8624H,0H
 7040 CURSOR 28,W: PRINT P;
 7050 CURSOR 1,16: PRINT "
 ";
 7052 FOR T=1 TO 10: NEXT T
 7054 CURSOR 1,16: PRINT "ダイセイコ
ウ!";
 7056 FOR T=1 TO 10: NEXT T
 7058 NEXT W
 7160 GOTO 6000
> _
```

50手先まで
詰将棋解読プログラム

TK-80BS機械語

井上政夫

私は以前から「考えるマイコン」を作ってみたいと思っていました。オセロゲームなどはいい例です。人間相手にマイコンがオセロをやるなんてまさに「考えるマイコン」だと思います。そこで私もと思い，何かいい題材はないかと考えた末マイコンに詰将棋をやらせてみましたので，ここで紹介させていただきます。ただ処理時間が長くかかるという点でそのまま実用に向くようなプログラムではないかもしれません。しかし，詰将棋を解いていくアルゴリズムなど興味深いものがあり，アイデア次第でもっとすぐれたプログラムが出来ると思います。この点で私のプログラムが一種の踏み台になれば幸いです。

使用したマイコンの機種はTK-80BSでRAMは7KB実装ですが5KBの標準実装でもプログラムを一か所書き換えればOKです。ハードの改造はしていません。プログラム言語は機械語で2.5KB弱のプログラムとなりました。

サブルーチン群について

まず詰将棋のプログラムの前提となるいくつかのサブルーチンについて説明します。

今回紹介するプログラムは大きく二つに分かれます。そのひとつは詰将棋のアルゴリズムにかかわるもので，もうひとつはここで述べる2KB弱のプログラムから成るサブルーチン群です。その仕様は**第31－1図**に示します。これらは詰将棋の一手一手の指し手を捜し出してくるものですが次にこれらのことを順を追って説明します。

攻め手に関するサブルーチン

まず詰将棋をやろうというからには将棋盤と駒台が必要で，それらは**第31－2図**に示すようにメモリ上に構成されます。ここがサブルーチン群のワーキング・エリアになるのです。

まず持ち駒についてはそれぞれ指定されたメモリにその駒の数を書き込みます。攻め方に持ち駒がなければ当然8202～8208番地は0とします。

またここで注意しなければいけないのは玉方の持ち駒も忘れずセットするということです。次に盤面の方ですがそこには駒を置かなければなりません。**第31－3図**を見て下さい。これは盤を構成するメモリの1バイト1バイトのビット構成です。下位4ビットは駒コードを入れます。第5ビットはその駒が攻め方のものか玉方のものかを決めるもので玉方の駒なら1，攻め方の駒かあるいは駒が置いてないなら0です。

上位3ビットは駒のききに関するものでこれはプログラム中で処理されますから駒をセットするときはどうなっていてもかまいません。駒コードについては**第31－4図**を見て下さい。

ここで9というのは未定義なので使わないで下さい。また攻め方の玉，たとえば01Hなどを入れてはこまります。これらの駒のセット以外にもうひ

ラベル	名　　称	アドレス	入力レジスタ	出力レジスタ	保存	機　　能
CLR	クリアー	8E00	なし	なし	なし	(8200)～(82AF)＝00玉方の持ち駒，玉アドレスのセット。
SMI	攻め方-INITIALIZE	8F30	なし	A～E	H,L	攻め方の手に関するデータを初期化し，レジスタにセット。
SMA	攻め方-AHEAD	8832	SMIまたはSMCの出力	A～EおよびH,L	なし	攻め方の手を一手すすめデータをセット。手がないならA＝0。
SMB	攻め方-BACK	8B98	SMAの出力(H,Lは除く)	なし	A～L	入力データに従って攻め方の手を一手もどす。
SMC	攻め方-CHANGE	8F3B	SMAの出力(H,Lは除く)	A～E	H,L	攻め方の手のデータを書き替える。
SMD	攻め方-DECODE	8E82	SMAの出力HL＝V.R.A	なし	なし	攻め方の手のデータを解読し，TVに表示。
GKI	玉方-INITIALIZE	8F72	SMAの出力	A～E	なし	玉方の手についてSMIと同様。
GKA	玉方-AHEAD	8917	GKIまたはGKCの出力	A～E	なし	〃　SMA　〃
GKB	玉方-BACK	8BB8	GKAの出力	なし	A～L	〃　SMB　〃
GKC	玉方-CHAGE	8FB1	GKAの出力	A～E	H,L	〃　SMC　〃
GKD	玉方-DECODE	8E92	GKAの出力,HL＝V.R.A	なし	なし	〃　SMD　〃
DPA	DISPLAY-A	8F15	なし	なし	なし	"ツミナシ"とTVに表示。
DPB	DISPLAY-B	8F2A	なし	なし	なし	"メモリオーバーフロー"とTVに表示。

注意 V.R.Aはビデオ RAMアドレスを示す。「保存」とは内容が書き変わらないレジスタを示す。
A～EはA,B,C,D,Eのレジスタを示し，A～LはA～E，H，Lのレジスタを示す。
以上のサブルーチンは8832～8FD7番地のプログラムを必要とし，さらに8200～82AF番地はワーキングエリアとなる。

《第31―1図》サブルーチン群

《第31―2図》ワーキングエリア将棋盤と駒台

《第31―3図》盤のビット構成

```
@
*GO,8600
#8E00
*CM,8208
     8208  00-01
     8209  00-
*CM,8220
     8220  00-17
     8221  00-00
     8222  00-11
     8223  00-16
     8224  00-
*CM,8232
     8232  00-03_
```

《写真31―1》データの入力

無	玉	飛	角	銀	桂	香	歩
0	1	2	3	4	5	6	7
金	不定	竜	馬	成り銀	成り桂	成り香	と
8	9	A	B	C	D	E	F

《第31―4図》駒コード

とつ玉のアドレスは特別に**第31―2図**に示す場所に格納されていなければなりません。以上で駒のセットは終わりです。

この駒のセットを助けるサブルーチンがCLRです。

これをコールすると盤面や駒台を0クリアした後，RST命令でBSモニタに飛びます。そこでCMコマンドで盤面の駒組みと攻め方の持ち駒だけ入力します。その後RTコマンドでプログラムを再スタートすると玉方の持ち駒と玉アドレスは自動的にセットされます。

これでおぜん立てができました。そこで次に攻め方の指し手に関するサブルーチンについて述べます。まずSMIをコールするとA, B, C, D, Eのレジスタ（以後A～Eと略記）に攻め方の手に関するデータを初期化してセットしもどってきます。

ラベル	ニーモニック
START	CALL CLR
	CALL SMI
	CALL SMA
	CPI 00
	JZ STOP
	PUSH PSW
	PUSH B
	PUSH D
	LXI H, 7E00
	CALL SMD
	POP D
	POP B
	POP PSW
	CALL SMB
	CALL SMC
	CALL SMA
	CPI 00
	JZ STOP
	LXI H, 7E20
	CALL SMD
STOP	HLT

《第31—5図》サブルーチン応用例

4	3	2	1	
			歩	一
				二
		角	王	三
	角		香	四
	歩			五
				六

持駒　金

《第31—6図》詰将棋例題

《写真31－2》結果表示

アドレス	コード
8208	01
8220	17
8222	11
8223	16
8232	03
8243	03
8244	07

《第31—7図》駒入力例

␣␣␣␣1－二␣キン␣␣ウチ
␣␣␣␣2－四␣キン␣␣ウチ
2－三␣1－二␣カク␣␣ナリ
注. ␣は空白

《第31—8図》指し手表示例

それをそのままSMAに渡すと盤面や持ち駒を調べた上で王手をひとつ見つけ，その手を指し，指し手のデータをA～Eにセットしてもどってきます。これで王手が一つ見つかったのですが，その手以外にもいろいろな王手があるはずです。それを見つけるにはまずさっきのデータA～EをSMBに渡します。SMBはデータには一切手を加えずただその手を読んで指し手をもどします。

これで局面は最初の状態になりました。次にデータをSMCに渡します。SMCはデータの内容を一部書き換えます。これをSMAに渡すと今度はさっきとは異なった王手を見つけてきてくれます。同様にこのデータをSMB，SMCを通してSMAに送ればまた別の手を捜してきます。しかしこれを繰り返していけばそのうち手がなくなります。あるいは将棋の局面によっては始めから王手ができない場合があります。SMAでは手を捜がし，これ以上指し手がない場合にはAレジスタに0を入れてもどってきます。

よってSMAをコールした後は必ずAレジスタをチェックします。以上が攻め方の手を見つける方法ですが実際にSMAはどんな王手を指したのかを知るにはSMAの出力データA～Eを解読するか，**第31—2図**の盤面を調べて駒がどのように動かされたかを見なければなりません。これではたいへんですのでそのためのデコーダとしてSMDがあります。SMAの出力データA～Eと，H，L，レジスタにビデオRAMのアドレス（例えば7E00）を入れてSMDをコールするとTV画面に指し手を表示してくれます。

以上が攻め方の手に関するサブルーチンの説明ですが，これらの具体的応用例を**第31—5図**に示します。このプログラムを追ってみますとまずCLRをコールしているのでさっき述べたようにブレイクしてBSモニターにもどってきます。ここで駒のセットを行なうのですが，例えば**第31—6図**に示す詰将棋を入力するとします。それにはCMコマンドで**第31—7図**のようにメモリに書き込めばよいことは今までの説明でわかってもらえると思います。

駒の入力後，RTコマンドを行なうと（**第31—5図**にもどって）SMI，SMAを通って一応王手があったか確認した後データを退避し，SMDでその指し手をTVに表示します。SMDではデータはこわされてしまいますのでPOP命令でデータを復帰させてSMB，SMCに送り，SMAで他の王手を指させ，それをまたSMDで表示します。このプログラムを実行すると**第31—8図**の上の2行に示

すように指し手がTVに表示されます。**第31－5図**のプログラムをさらに拡張して三つ目の手を表示させると**第31－8図**の3行目のようになります。

この図から指し手の表示の形式がわかってもらえると思います。一応ことわっておきますが，盤面の駒の移動による手の表示には，移動前の位置（**第31－8図**の例では2—三）と移動後の位置（**第31－8図**の3行目の1—二）の両方が示されます。

玉方の手

攻め方の手に関する説明が終わりましたので次には玉方の手ですが，**第31－1図**を見るとわかるように攻め方のサブルーチンに対応して玉方にもGKI, GKA, GKB, GKC, GKDのサブルーチンがあり，ほとんど同じ役目をします。ただGKAが捜してくる指し手は王手ではなく，玉方の逃げる手です。また一つ注意を要する点はSMIでは何の入力も必要なかったのですが，GKIではSMAの出力すべてが入力として必要です。特にSMAは指し手のデータA～Eの他にGKIでのみ必要なデータをH, Lのレジスタにセットしてきますから，それらをこわさずGKIに渡してやらなければなりません。

これで主要なサブルーチンは出そろいましたが**第31－1図**をみるとまだDPAとDPBが残っています。しかし，これらは詰将棋のプログラムを作ったときにちょっと便利だと思って作っただけのものです。

ただBSモニタ・プログラムのサブルーチンを利用しているので，プログラムの内容を調べてみるのも何かの参考になるかも知れません。

以上で13個のサブルーチンが出てきましたが，これらは8832～8FD7番地2KB弱のプログラムで動きます。このプログラムの詳細についてはここでは省略させてもらいます。また以上のプログラムがRAMに書き込まれている時はスタックポインタSPに注意して下さい。BSモニターのGOコマンドを使用するとSP＝8C00となっています。このままですと数レベル程度のスタックは問題ないのですが，あまりスタックを用いるとプログラムがこわされます。SPは適当な場所に書き換えるように心掛けて下さい。

サブルーチンの説明の総括として最後にもう一つの応用例を**第31－9図**に示します。その内容については各自で確認してみて下さい。結果だけ述べますとなにかの詰将棋の問題について**第31－9図**のプログラムを実行するとその処理が終わった時点でスタックにはひとつの詰め手順が格納されていることになります。しかし，これがその詰将棋の正解手順であるという保証はありません。

詰将棋のアルゴリズム

ではいよいよ核心にはいります。いままで説明してきたサブルーチンを利用して詰将棋をするプログラムを作るのですがその方法にはいろいろなものがあると思います。私自身いくつかの案をだしてみましたが，ここではその一つを紹介します。

それにはいくつかのパラメータを利用しているのでそれらを**第31－10図**に示します。まず指し手の格納場所を見て下さい。9000番地以下を256バ

ラベル		
START	CALL	CLR
LOOP	CALL	SMI
AGAIN	CALL	SMA
	CPI	00
	JZ	BACK
	PUSH	PSW
	PUSH	B
	PUSH	D
	CALL	GKI
	CALL	GKA
	CPI	00
	JZ	END
	PUSH	PSW
	PUSH	B
	PUSH	D
	JMP	LOOP
BACK	POP	D
	POP	B
	POP	PSW
	CALL	GKB
	POP	D
	POP	B
	POP	PSW
	CALL	SMB
	CALL	SMC
	JMP	AGAIN
END	HLT	

《第31－9図》サブルーチン応用例

《第31—10図》パラメータ，x，y，By，Ty，P，Mp

手　数	1 3 5 7 9 11 13 15……49	51
行座標	02 0C 16 20 2A 34 3E 48……F2	FC

《第33—11図》

《第31—12図》 分岐

イトごとに区切って一列づつ横に並べたものです。列座標yはアドレスの上位バイトで，行座標xは下位バイトで示します。そして一つの列には一つの詰め手順がはいります。図からわかるように行座標02からA～Eのレジスタを格納し，それが攻め方の手，玉方の手と交互に連なっていきます。各列の上の方にあるTyというのはその列に格納されている詰め手順の手数を示します。ただ実際の手数ではなく，最後の手が格納されている行座標の一番上，すなわち1手詰なら02，3手詰なら0Cとするのです。その対応表を**第31—11図**に示します。これからわかるように一列には最高49手詰まではいります。

分岐

また各列の最上にあるByというのはその列の指し手は左隣の列に格納されている詰め手順の何手目から分岐したものかを示します。この分岐ということについては**第31—12図**を見て下さい。これは指し手の格納を簡単化して示したもので，○は攻め方の手を，□は玉方の手を表わします。**第31—12図**では，一列目に3手詰の手順が格納されたところです。ここで3手目についてはその局面における攻め方の最善手であることは問題ありませんが，2手目の玉方の手は最善手とは限りません。他の手を指せば逃げきるかも知れないのです。

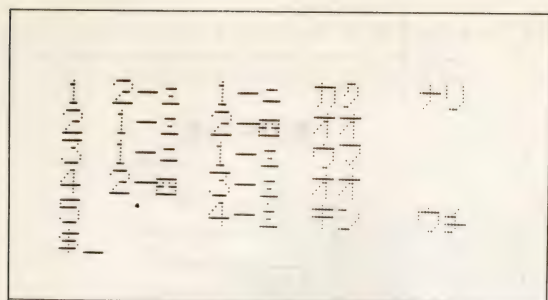
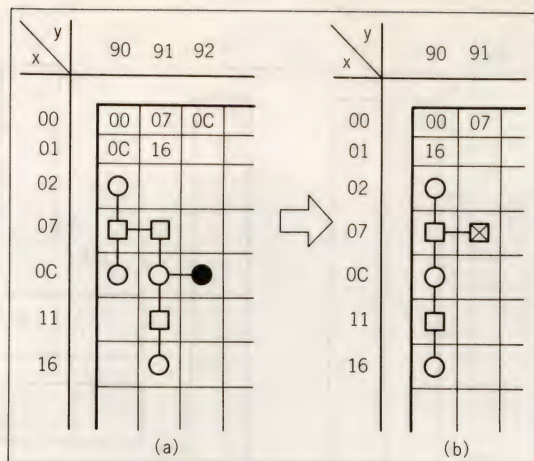
《写真33—3》入力例

そこで隣の列に2手目以降違う手を指してみるのです。これが分岐で**第31—12図**はその様子を示したものです。分岐するのは玉方の手だけでなく，攻め方の手でも分岐します。それぞれの分岐の目的は次のようなものです。

攻め方の分岐—今まで見つかった詰手順よりさらに短い手数の詰手順を見つける。

玉方の分岐—逃げ切る手を見つけるか，さもなければ手数のかかる詰手順にする。

このような目的のために次々に分岐し，指し手の格納場所は右の列へとずれていきます。言い忘れましたが，Byについても Tyと同様行座標を用います。また第一列目は分岐というものは考えら

```
1 2-三 1-二 カク  ナリ
2 1-三 2-四 オオ
3 1-二 1-三 ウマ
4 2-四 3-三 オオ
5  ・  4-三 キン  ウチ
キ_
```

《写真31―3》結果表示例

《第31―13図》収束

れませんので，$B_{90}=0$とします。

ところで攻め方の分岐の目的からして攻め方の手で分岐した以降は分岐前の詰手数が手数の上限となります。すなわち分岐後はその上限を越えるような詰手順が見つかっても無駄ですので，その上限を越えて手を読ませるのをやめさせます。そして，この上限は攻め方の手で分岐するごとに変わります。そこで，この上限を格納するのにプッシュダウンスタックのようなものを考えました。それは**第31―10図**に示してありますが，82B0番地から攻め方が分岐するごとに次々に分岐する手前の列の詰手数Tyを格納します。そしてそのスタックポインタをPで表わし，Pの指す所に格納されている上限をM_Pで表わすことにします。ただ一番最初の82B0番地にはあらかじめFCと書き込んでおきます。

第31―11図を見れば明らかなように，これは上限を51手詰とすることです。これは当然で一つの列には49手詰までしか格納できないからです。よって仮に51手詰以上の詰将棋を私のプログラムにやらせると，原理上“ツミナシ”と出てくることになります。

以上でパラメータが出そろいました。さて前に分岐ということを述べましたが，それに対して収束があります。その収束の様子を示したのが**第31―13図**です。まず (a) は2列目の3手目で分岐して3列目に移ったところです。ところが調べてみるとそれ以上他に手がなかったのです。●は攻め方の手がもうないことを示します。そこで前の列にもどってみると2手目で分岐してきていることがわかります。すなわち1列目と2列目に2手目以降が異なる詰手順がはいっています。そこでそのどちらかを選らぶのですが，この場合玉方の手で分岐しています。よって手数の方を選びます。すなわち2列目の手順を採用し，それを1列目に移すのです。これが収束で，収束した後の様子が**第31―13図**の (b) です。**第31―13図** ではさらに2手目で分岐し，⊠以降また手を続けようとしているのです。この例は玉方の手の分岐に対する収束の場合ですが，攻め方の手の分岐における収束は逆に手数の短い方を採用します。

ここで注意してもらいたいのは，さきほど手数の上限M_Pというのを決めました。よって攻め方の分岐点では絶えず右の列の方に短い詰手順がはいっているのです。

このため攻め方の収束では手数の比較をすることなく無条件に右の列の詰手順を採用し，それを左の列に移せばよいのです。

ここでプログラムの流れを大きく捕えると，まず指し手の格納場所の第1列目に次々に指し手を書き込み，とにかく一つの詰手順を見つけます。これはちょうど**第31―9図**に示した応用例のようなものです。詰手順がみつかると手をすこしもどし，必要な所は分岐していきます。そして分岐した先でまた詰手順が見つかると，さらに分岐します。次々に分岐し，指し手の末端まで調べるとこんどは収束します。収束してはもうすこし手前の手から分岐します。この分岐，収束を繰り返し，これ以上分岐できないとこまできてプログラムの処理は終了です。そしてこの時第1列目に正確手順

詰将棋フローチャート

NGK

By=x

No: NGA

LOD

SMB

打歩詰め？

Yes: SMC

No: Ty=x

By=x

Yes: MOV / y=y−1 / P=P−1

x=02

Yes: TMI

No: LOD / GKB / GKC

By=x

Yes

No: y=9F

Yes: IMP

No: y=y+1 / By=x

Yes: NGB

y=y−1

LOD

SMB

SMC

By=x

Yes: MOV / MP=Ty

No: P=P+1 / MP=Ty / y=y+1 / By=x

STR

指し手のデータ5バイトをストアー

x=x+5

RET

LOD

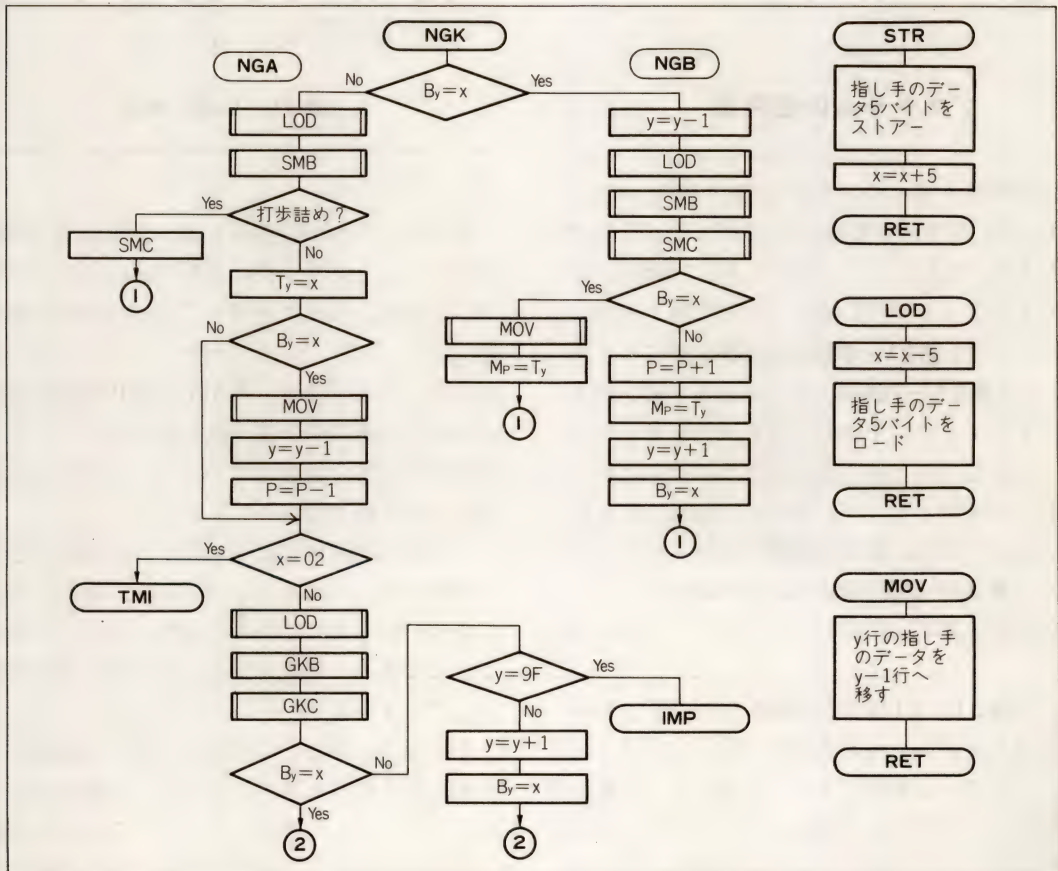
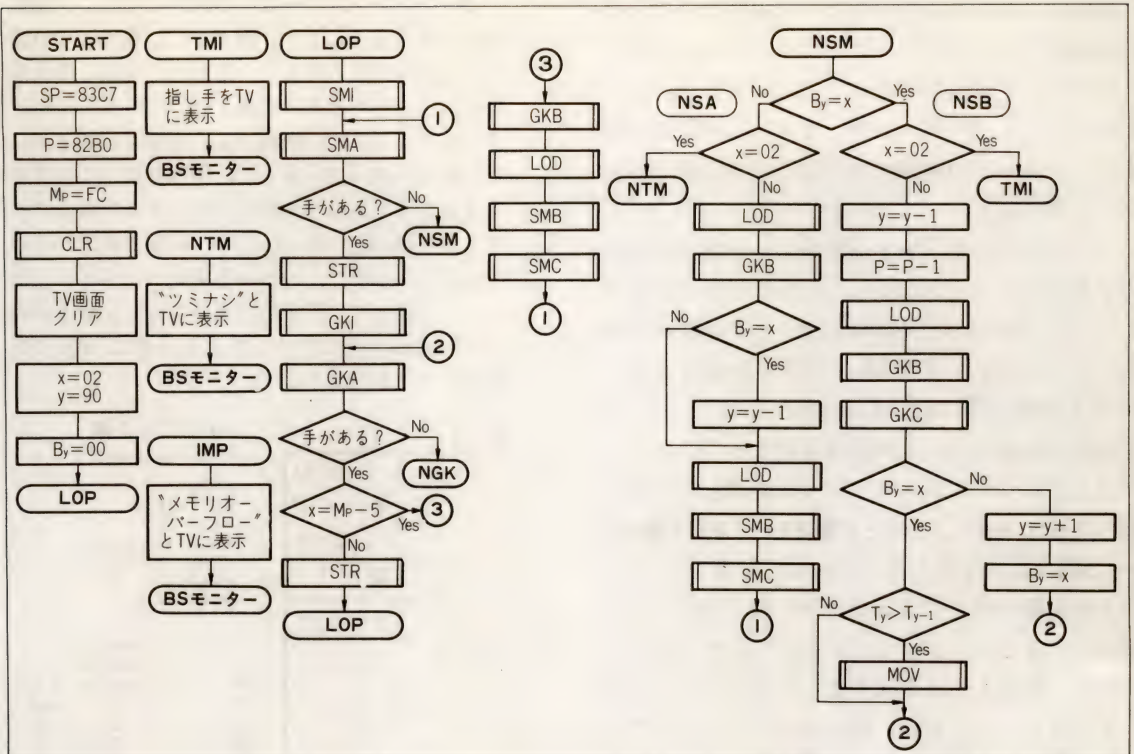
x=x−5

指し手のデータ5バイトをロード

RET

MOV

y行の指し手のデータをy−1行へ移す

RET

が格納されているのです。

以上はプログラムの漠然とした説明なので，その詳細はフローチャートにして示してみました。これをつぶさに説明するわけにはいきませんが，そのポイントだけ述べます。まずSTARTから始まりパラメータの初期化を行なった後LOP（ループ）に飛びます。

ここで次々に指し手を進めます。そして攻め方の手がなくなるとNSMへ，玉方の手がなくなるとNGKへ飛びます。NSMではそこが分岐点であるかどうかによってNSAとNSBに分かれます。NGKも同様にNGAとNGBに分かれます。これ以下分岐や収束についての処理を行なってまたループにもどるのですが，どのように処理しているか調べてみて下さい。このとき**第31－12図**や**第31－13図**のような図を書いてみるのも手です。

また実際のプログラムは8600～8815番地の約0.5KBで，なるべくわかりやすいように作ったつもりです。備考もつけておきましたのでフローチャートと見くらべてもいいかと思います。今まで述べてきたことを想起しつつ自分で考えてみて下さい。

プログラムの使用法

一応使用法を述べておきます。まずBSモニターのGOコマンドで8600番地からプログラムをスタートします。するとすぐにブレイクしてBSモニターにもどってくるのでCMコマンドで駒を入力します。ここでも**第31－6図**の詰将棋を例にとります。よって**第31－7図**のようにメモリを書き換えます。そしてさらに手数の上限を入力してください。これをしないと前に述べたように上限はFC，すなわち51手詰となっているので，無駄な手を数十手も先まで読み，非常に時間がかかります。実を言うと**第31－6図**の詰将棋は5手詰です。よってこの場合上限は7手詰にするとちょうどよいのです。

そこで**第31－11図**から82B 0 番地をFCから20に書き換えます。これで準備はOK，後はRTコマンドでプログラムを再スタートさせます。するとTV画面がクリアされ，6秒ほどたつと指し手が表示され，BSモニターのコマンド入力待ちになります。

次にいくつかの注意点を述べます。まず詰将棋の問題によっては，特に手数の長いものはプログラム処理中，次々に分岐してついには指し手を格納できなくなり“メモリーオーバーフロー”と表示されてしまいます。RAMが5KB標準実装の場合はさらにその可能性が高くなります。ところでRAMが5KBの場合はプログラムの8730番地を97Hと書き換えておいて下さい。

また普通詰将棋では最後に玉方が合駒をしてもその駒を取って詰みという場合は，合駒をする前の時点で詰んだと認められます。しかしプログラムではなるべく手数を長くしようとするので合駒をしてしまいます。

また開き王手などに対しても次々と合駒をして手数が長くなります。よって正しく駒を入力したのに“ツミナシ”と出てきたら，こんどは上限をもうすこし高くしてみて下さい。私の経験したも

このプログラムは一般の市販書の解説より実戦向きと言えるでしょう。例として，**第31－14図**のような場合を考えます。これは，八段，米長邦雄氏の「これが米長詰将棋」(昭文社)より引用しています。米長氏の解説では，最短コースを選んで三手詰となっていますが，同様な事をこのプログラムでは五手詰と判断します。これは別にプログラムの頭が悪いのではなく，相手の攻撃を十分に考慮したもので，いわゆる実戦タイプと言えましょう。従って，三手詰の場合でも，上限を高めに7手詰においたほうが良いという事は，記事に説明されているとおりです。

《第31－14図》実例

のでは3手詰と書いてある詰将棋が結局7手詰になって解けたことがあります。

それから注意点ではありませんがプログラムの処理時間について述べます。これは一概には言えないのですが，上限を正しくセットした場合，3手詰で数秒から十秒前後，5手詰で十秒から数分，7手詰で5分前後から数十分，9手詰になると1時間を越えるものがでそうです。

詰将棋の正解はひとつとは限らずいろいろな変化手順をもっているのが普通です。私のプログラムではその中の一つしか示してくれないのが不満です。こういった欠点はまだあるのですが，そのようなことよりもまず処理時間を短くすることが先決問題です。それには無駄な手をなるべく読まないですむ，効率のよいアルゴリズムを見つけることだと思います。

改良詰将棋プログラム

アドレス	機械語	ラベル	ニーモニック	備考
8600	31 C7 83	START	LXI SP,83C7	SP=83C7
3	3E FC		MVI A,FC	Mp=FC
5	32 B0 82		STA 82B0	
8	CD 00 8E		CALL CLR	
B	CD 6C FA		CALL FA6C	TV画面クリア
E	21 B0 82		LXI H,82B0	P=82B0
11	22 C0 82		SHLD 82C0	
4	21 00 90		LXI H,9000	
7	36 00		MVI M,00	By=00
9	2E 02		MVI L,02	
B	22 C2 82		SHLD 82C2	x=02 y=90
E	C3 36 86		JMP LOP	
8621	21 01 90	TMI	LXI H,9001	L=x H=y
4	CD 7E 87		CALL DIP	
7	C3 42 F0		JMP F042	
862A	CD 15 8F	NTM	CALL DPA	
D	C3 42 F0		JMP F042	
8630	CD 2A 8F	IMP	CALL DPB	
3	C3 42 F0		JMP F042	
8636	CD 30 8F	LOP	CALL SMI	
8639	CD 32 88	LOP-1	CALL SMA	
C	A7		ANA A	A=0?
D	CA 74 86		JZ NSM	
40	E5		PUSH H	
1	CD C8 87		CALL STR	
4	E1		POP H	
5	CD 72 8F		CALL GKI	
86.48	CD 17 89	LOP-2	CALL GKA	
B	A7		ANA A	A=0?
C	CA D8 86		JZ NGK	
F	F5		PUSH PSW	
50	2A C0 82		LHLD 82C0	HL=P
3	7E		MOV A,M	A=Mp
4	D6 05		SUI 05	A=Mp-5
6	21 C2 82		LXI H,82C2	M=x
9	BE		CMP M	x=Mp-5?
A	CA 64 86		JZ LOP-3	
D	F1		POP PSW	
E	CD C8 87		CALL STR	
61	C3 36 86		JMP LOP	
8664	F1	LOP-3	POP PSW	
5	CD B8 8B		CALL GKB	
8	CD D9 87		CALL LOD	
B	CD 98 8B		CALL SMB	
E	CD 3B 8F		CALL SMC	
71	C3 39 86		JMP LOP-1	
8674	2A C2 82	NSM	LHLD 82C2	L=x H=y
7	45		MOV B,L	B x
8	2E 00		MVI L,00	
A	78		MOV A,B	A=x
B	BE		CMP M	B=x
C	CA A0 86		JZ NSB	
867F	FE 02	NSA	CPI 02	x=02?
81	CA 2A 86		JZ NTM	
4	CD D9 87		CALL LOD	L=x H=y
7	CD B8 8B		CALL GKB	
A	45		MOV B,L	B=x
B	2E 00		MVI L,00	M=By
D	7E		MOV A,M	A=By
E	B8		CMP B	By=x?
F	C2 93 86		JNZ NSA-1	
92	25		DCR H	y=y-1
8693	68	NSA-1	MOV L,B	
4	CD DC 87		CALL LODX	
7	CD 98 8B		CALL SMB	
A	CD 3B 8F		CALL SMC	
D	C3 39 86		JMP LOP-1	
86A0	FE 02	NSB	CPI 02	x=02?
2	CA 21 86		JZ TMI	
5	25		DCR H	y=y-1
6	4C		MOV C,H	C=y
7	21 C0 82		LXI H,82C0	M=P
A	35		DCR M	P=P-1
B	61		MOV H,C	H=y
C	68		MOV L,B	L=x
D	CD DC 87		CALL LODX	L=x H=y
B0	CD B8 8B		CALL GKB	
3	CD B1 8F		CALL GKC	
6	F5		PUSH PSW	指し手のデータを退避
7	C5		PUSH B	
8	45		MOV B,L	B=x
9	2E 00		MVI L,00	M=By
B	7E		MOV A,M	A=By
C	B8		CMP B	By=x?
D	C2 CD 86		JNZ NSB-1	
CO	2C		INR L	L=01
1	25		DCR H	M=Ty-1
2	7E		MOV A,M	A=Ty-1
3	24		INR H	M=Ty
4	BE		CMP M	Ty>Ty-1
5	DC EA 87		CC MOV	
8	C1		POP B	
9	F1		POP PSW	
A	C3 48 86		JMP LOP-2	
86CD	24	NSB-1	INR H	y=y+1
E	70		MOV M,B	By=x
F	7C		MOV A,H	
D0	32 C3 82		STA 82C3	yを格納
3	C1		POP B	
4	F1		POP PSW	
5	C3 48 86		JMP LOP-2	
86D8	2A C2 82	NGK	LHLD 82C2	L=x H=y
B	45		MOV B,L	B=x
C	2E 00		MVI L,00	M=By
E	7E		MOV A,M	A=By
F	B8		CMP B	By=x?
E0	CA 44 87		JZ NGB	
86E3	CD D9 87	NGA	CALL LOD	L=x H=y
6	CD 98 8B		CALL SMB	
9	3D		DCR A	打歩詰めかどうかを
A	C2 FB 86		JNZ NGA-1	調べます。それには
D	78		MOV A,B	A=01?
E	FE 07		CPI 07	B=07?
F0	C2 FB 86		JNZ NGA-1	をチェックします
3	3E 01		MVI A,01	
5	CD 3B 8F		CALL SMC	
8	C3 39 86		JMP LOP-1	
86FB	45	NGA-1	MOV B,L	B=x
C	2E 01		MVI L,01	
E	70		MOV M,B	Ty=x
F	2D		DCR L	M=By
700	78		MOV A,B	A=x
1	BE		CMP M	By=x?
2	C2 10 87		JNZ NGA-2	

アドレス	機械語	ラベル	ニーモニック		備考
5	CD EA 87		CALL	MOV	
8	25		DCR	H	y=y-1
9	4C		MOV	C,H	c=y
A	21 C0 82		LXI	H,82C0	M=P
D	35		DCR	M	P=P-1
E	61		MOV	H,C	H=y
F	78		MOV	A,B	A=x
8710	FE 02	NGA-2	CPI	02	x=02?
2	CA 21 86		JZ	TMI	
5	68		MOV	L,B	L=x
6	CD DC 87		CALL	LODX	L=x H=y
9	CD B8 8B		CALL	GKB	
C	CD B1 8F		CALL	GKC	
F	F5		PUSH	PSW	} 指し手のデータ
20	C5		PUSH	B	} 退避
1	45		MOV	B,L	
2	2E 00		MVI	L,00	
4	7E		MOV	A,M	} By=x ?
5	B8		CMP	B	
6	C2 2E 87		JNZ	NGA-3	
9	C1		POP	B	
A	F1		POP	PSW	
B	C3 48 86		JMP	LOP-2	
872E	7C	NGA-3	MOV	A,H	} y=9F ?
F	FE 9F		CPI	9F	
31	C2 39 87		JNZ	NGA-4	
4	C1		POP	B	
5	F1		POP	PSW	
6	C3 30 86		JMP	IMP	
8739	24	NGA-4	INR	H	y=y+1
A	70		MOV	M,B	By=x
B	7C		MOV	A,H	} yを格納
C	32 C3 82		STA	82C3	
F	C1		POP	B	
0	F1		POP	PSW	
41	C3 48 86		JMP	LOP-2	
8744	25	NGB	DCR	H	y=y-1
5	68		MOV	L,B	
6	CD DC 87		CALL	LODX	L=x H=y
9	CD 98 8B		CALL	SMB	
C	CD 3B 8F		CALL	SMC	
F	F5		PUSH	PSW	} 指し手のデータ
50	C5		PUSH	B	} を退避
1	45		MOV	B,L	
2	2E 00		MVI	L,00	
4	7E		MOV	A,M	} By=x ?
5	B8		CMP	B	
6	C2 68 87		JNZ	NGB-1	
9	CD EA 87		CALL	MOV	
C	2E 01		MVI	L,01	M=Ty
E	7E		MOV	A,M	A=Ty
F	2A C0 82		LHLD	82C0	HL=P
62	77		MOV	M,A	Mp=Ty
3	C1		POP	B	
4	F1		POP	PSW	
5	C3 39 86		JMP	LOP-1	
8768	2C	NGB-1	INR	L	M=Ty
9	4E		MOV	C,M	C=Ty
A	24		INR	H	y=y+1
B	2D		DCR	L	
C	70		MOV	M,B	By=x
D	7C		MOV	A,H	} yを格納
E	32 C3 82		STA	82C3	
71	2A C0 82		LHLD	82C0	HL=P
4	2C		INR	L	P=P+1
5	71		MOV	M,C	Mp=Ty
6	22 C0 82		SHLD	82C0	Pを格納
9	C1		POP	B	
A	F1		POP	PSW	
B	C3 39 86		JMP	LOP-1	
877E	7E	DIP	MOV	A,M	A=Ty
F	23		INX	H	
80	22 C2 82		SHLD	82C2	x,yを格納
3	C6 05		ADI	05	} B=Ty+5
5	47		MOV	B,A	
6	0E 31		MVI	C,31	C=31("1"のキャ
8788	C5	DIP-1	PUSH	B	ラクターコード)
9	2A 7F 84		LHLD	847F	HL=ビデオRAMアド
C	71		MOV	M,C	レス何手目かを表示

アドレス	機械語	ラベル	ニーモニック		備考
D	23		INX	H	
H	23		INX	H	
F	E5		PUSH	H	
90	CD 05 88		CALL	LOAD	攻め方の手をロード
3	E1		POP	H	HL=ビデオRAMアド
4	CD 82 8E		CALL	SMD	レス攻め方の手を表示
7	21 7E 84		LXI	H,847E	M=Y(TV画面の縦座標)
A	34		INR	M	Y=Y+1
B	CD B9 FA		CALL	FAB9	ビデオRAMアドレスセット
E	C1		POP	B	
F	3A C2 82		LDA	82C2	} x=Ty+5 ?
A2	B8		CMP	B	
3	C8		RZ		
4	79		MOV	A,C	
5	FE 39		CPI	39	39は"9"のキャラクターコ
7	C8		RZ		ード9手目まで表示
8	0C		INR	C	
9	C5		PUSH	B	HL=ビデオRAMアド
A	2A 7F 84		LHLD	847F	レス何手目かを表示
D	71		MOV	M,C	
E	23		INX	H	
F	23		INX	H	
B0	E5		PUSH	H	
1	CD 05 88		CALL	LOAD	玉方の手をロード
4	E1		POP	H	
5	CD 92 8E		CALL	GKD	玉方の手を表示
8	21 7E 84		LXI	H,847E	
B	34		INR	M	
C	CD B9 FA		CALL	FAB9	
F	C1		POP	B	
C0	0C		INR	C	
1	C3 88 87		JMP	DIP-1	
4	00 00		NOP	2	
6	00 00		NOP	2	
87C8	2A C2 82	STR	LHLD	82C2	L=x H=y
87CB	77	STRX	MOV	M,A	
C	23		INX	H	
D	70		MOV	M,B	
E	23		INX	H	
F	71		MOV	M,C	
D0	23		INX	H	} 指し手をストアー
1	72		MOV	M,D	
2	23		INX	H	
3	73		MOV	M,E	
4	23		INX	H	
5	22 C2 82		SHLD	82C2	x=x+5 y=y
8	C9		RET		
87D9	2A C2 82	LOD	LHLD	82C2	L=x H=y
87DC	2B	LODX	DCX	H	
D	5E		MOV	E,M	
E	2B		DCX	H	
F	56		MOV	D,M	
E0	2B		DCX	H	
1	4E		MOV	C,M	} 指し手をロード
2	2B		DCX	H	
3	46		MOV	B,M	
4	2B		DCX	H	
5	7E		MOV	A,M	
6	22 C2 82		SHLD	82C2	x=x-5 y=y
9	C9		RET		
87EA	D5	MOV	PUSH	D	指し手のデータを退避
B	54		MOV	D,H	
C	15		DCR	D	D=y-1
D	2E 01		MVI	L,01	
F	5D		MOV	E,L	
F0	7E		MOV	A,M	} Ty-1=Ty
1	12		STAX	D	
2	C6 04		ADI	04	} C=Ty+4
4	4F		MOV	C,A	
5	68		MOV	L,B	L=x
6	58		MOV	E,B	E=x
87F7	7E	MOV-1	MOV	A,M	y行のデータをy−1行へ。
8	12		STAX	D	(x,y−1)←(x,y)
9	79		MOV	A,C	(x+1, y−1)←
A	BD		CMP	L	(x+1, y)
B	CA 03 88		JZ	MOV-2	
E	2C		INR	L	(Ty+4, y−1)←
F	1C		INR	E	(Ty+4, y)
800	C3 F7 87		JMP	MOV-1	

アドレス	機械語	ラベル	ニーモニック	備考
8803	D1	MOV-2	POP D	
4	C9		RET	
8805	2A C2 82	LOAD	LHLD 82C2	LOADはサブルーチンDIP用の指し手をロードするサブルーチンです。
8	7E		MOV A,M	
9	23		INX H	
A	46		MOV B,M	
B	23		INX H	
C	4E		MOV C,M	
D	23		INX H	
E	56		MOV D,M	
F	23		INX H	
10	5E		MOV E,M	
1	23		INX H	
2	22 C2 82		SHLD 82C2	
5	C9		RET	
	8816〜8831は空です。			
8832	CD 00 8C	SMA	CALL KIK	
5	FE 01		CPI 01	
7	C2 77 88		JNZ KID	
A	C3 46 88		JMP KUC-3	
883D	0E 00	KUC	MVI C,00	
883F	78	KUC-1	MOV A,B	
40	FE 08		CPI 08	
2	CA 82 88		JZ KID-1	
8845	04	KUC-2	INR B	
8846	58	KUC-3	MOV E,B	
7	1A		LDAX D	
8	A7		ANA A	
9	CA 3F 88		JZ KUC-1	
C	C3 55 88		JMP KUC-5	
884F	79	KUC-4	MOV A,C	
50	E6 78		ANI 78	
2	C6 08		ADI 08	
4	4F		MOV C,A	
8855	78	KUC-5	MOV A,B	
6	CD A0 8C		CALL DRN	
9	FE FF		CPI FF	
B	CA 3D 88		JZ KUC	
E	4F		MOV C,A	
F	2A 10 82		LHLD 8210	
62	5C		MOV E,L	
3	CD 56 8D		CALL ADB	
6	E6 0F		ANI 0F	
8	C2 4F 88		JNZ KUC-4	
B	CD DD 8A		CALL NFU	
E	D2 45 88		JNC KUC-2	
71	3E 01		MVI A,01	
3	C9		RET	
4	00 00 00		NOP 3	
8877	6F	KID	MOV L,A	
8	79		MOV A,C	
9	E6 80		ANI 80	
B	CA 91 88		JZ KID-3	
E	62		MOV H,D	
F	C3 E6 88		JMP KID-9	
8882	1E 20	KID-1	MVI E,20	
8884	0E 00	KID-2	MVI C,00	
6	1A		LDAX D	
7	E6 1F		ANI 1F	
9	CA 9A 88		JZ KID-4	
C	E6 10		ANI 10	
E	C2 9A 88		JNZ KID-4	
8891	1A	KID-3	LDAX D	
2	CD A0 8C		CALL DRN	
5	FE FF		CPI FF	
7	C2 AE 88		JNZ KID-5	
889A	13	KID-4	INX D	
B	7B		MOV A,E	
C	E6 0F		ANI 0F	
E	FE 09		CPI 09	
A0	C2 84 88		JNZ KID-2	
3	7D		MOV A,E	
4	C6 07		ADI 07	
6	5F		MOV E,A	
7	FE B0		CPI B0	
9	C2 84 88		JNZ KID-2	
C	AF		XRA A	
D	C9		RET	

アドレス	機械語	ラベル	ニーモニック	備考
88AE	4F	KID-5	MOV C,A	
F	CD 31 8D		CALL ADA	
B2	FE F9		CPI F9	
4	CA C1 88		JZ KID-6	
7	E6 1F		ANI 1F	
9	CA CA 88		JZ KID-7	
C	E6 10		ANI 10	
E	C2 CA 88		JNZ KID-7	
88C1	79	KID-6	MOV A,C	
2	E6 78		ANI 78	
4	C6 08		ADI 08	
6	4F		MOV C,A	
7	C3 91 88		JMP KID-3	
88CA	46	KID-7	MOV B,M	
B	CD 0A 8B		CALL NRI	
E	A7		ANA A	
F	CA EE 88		JZ KID-10	
D2	3D		DCR A	
3	CA E2 88		JZ KID-8	
6	C5		PUSH B	
7	1A		LDAX D	
8	F5		PUSH PSW	
9	CD 07 8A		CALL OTE	
C	A7		ANA A	
D	CA 08 89		JZ KID-12	
E0	F1		POP PSW	
1	C1		POP B	
88E2	79	KID-8	MOV A,C	
3	F6 80		ORI 80	
5	4F		MOV C,A	
88E6	C5	KID-9	PUSH B	
7	1A		LDAX D	
8	F5		PUSH PSW	
9	F6 08		ORI 08	
B	C3 F1 88		JMP KID-11	
88EE	C5	KID-10	PUSH B	
F	1A		LDAX D	
F0	F5		PUSH PSW	
88F1	CD 07 8A	KID-11	CALL OTE	
4	A7		ANA A	
5	CA 08 89		JZ KID-12	
8	F1		POP PSW	
9	C1		POP B	
A	78		MOV A,B	
B	E6 0F		ANI 0F	
D	C2 C1 88		JNZ KID-6	
900	79		MOV A,C	
1	E6 7F		ANI 7F	
3	3C		INR A	
4	4F		MOV C,A	
5	C3 91 88		JMP KID-3	
8908	F1	KID-12	POP PSW	
9	57		WOV D,A	
A	7D		MOV A,L	
B	68		MOV L,B	
C	61		MOV H,C	
D	C1		POP B	
E	CD 43 8B		CALL INR	
11	CD 00 8C		CALL KIK	
4	C9		RET	
5	00 00		NOP 2	
8917	FE 01	GKA	CPI 01	
9	CA CD 89		JZ NGE-4	
C	D2 8C 89		JNC IGM-1	
F	79		MOV A,C	
20	C3 3A 89		JMP KTR-3	
8923	79	KTR	MOV A,C	
4	FE		CPI 40	
6	DA 31 89		UC KTR-1	
9	C2 72 89		JNZ IGM	
C	0E 48		MVI C,48	
E	C3 40 89	KTR-1	JMP KTR-4	
8931	0C		INR C	
2	C3 40 89		JMP KTR-4	
8935	79	KTR-2	MOV A,C	
6	E6 78		ANI 78	
8	C6 08		ADI 08	
893A	FE 50	KTR-4	CPI 50	
C	CA 72 89		JZ IGM	

アドレス	機械語	ラベル	ニーモニック		備考
F	4F		MOV	C,A	
8940	CD 31 8D	KTR-4	CALL	ADA	
3	FE F9		CPI	F9	
5	CA 35 89		JZ	KTR-2	
8	E6 1F		ANI	1F	
A	CA 23 89		JZ	KTR	
D	FE 12		CPI	12	
F	DA 35 89		JC	KTR-2	
52	CD A0 8C		CALL	DRN	
5	C2 35 89		JNZ	KTR-2	
8	EB		XCHG		
9	C5		PUSH	B	
A	CD 4A 8A		CALL	AOT	
D	C1		POP	B	
E	EB		XCHG		
F	A7		ANA	A	
60	CA 35 89		JZ	KTR-2	
3	7E		MOV	A,M	
4	12		STAX	D	
5	57		MOV	D,A	
6	36 00		MVI	M,00	
8	7D		MOV	A,L	
9	CD 43 8B		CALL	INR	
C	C9		RET		
D	00 00 00		NOP	3	
70	00 00		NOP	2	
8972	21 29 82	IGM	LXI	H,8229	
5	5D		MOV	E,L	
6	CD 4A 8A		CALL	AOT	
9	A7		ANA	A	
A	C2 BF 89		JNZ	NGE	
D	79		MOV	A,C	
E	E6 07		ANI	07	
80	CA BF 89		JZ	NGE	
3	79		MOV	A,C	
4	E6 78		ANI	78	
6	4F		MOV	C,A	
7	2E 12		MVI	L,12	
9	C3 9D 89		JMP	IGM-4	
898C	62	IGM-1	MOV	H,D	
D	68		MOV	L,B	
E	C3 9D 89		JMP	IGM-4	
8991	79	IGM-2	MOV	A,C	
2	E6 78		ANI	78	
4	4F		MOV	C,A	
5	68		MOV	L,B	
8996	7D	IGM-3	MOV	A,L	
7	FE 18		CPI	18	
9	CA BF 89		JZ	NGE	
C	23		INX	H	
899D	AF	IGM-4	XRA	A	
E	BE		CMP	M	
F	CA 96 89		JZ	IGM-3	
A2	45		MOV	B,L	
3	2A 10 82		LHLD	8210	
6	5D		MOV	E,L	
89A7	CD 56 8D	IGM-5	CALL	ADB	
A	E6 0F		ANI	0F	
C	C2 91 89		JNZ	IGM-2	
F	CD DD 8A		CALL	NFU	
B2	DA B9 89		JC	IGM-6	
5	0C		INR	C	
6	C3 A7 89		JMP	IGM-5	
89B9	3E 02	IGM-6	MVI	A,02	
B	C9		RET		
C	00 00 00		NOP	3	
89BF	0E 00	NGE	MVI	C,00	
C1	2A 10 82		LHLD	8210	
4	5D		MOV	E,L	
5	C3 D5 89		JMP	NGE-2	
89C8	79	NGE-1	MOV	A,C	
9	C6 08		ADI	08	
B	4F		MOV	C,A	
C	00		NOP		
D	79	NGE-4	MOV	A,C	
E	FE 40		CPI	40	
D0	C2 D5 89		JNZ	NGE-2	
3	AF		XRA	A	
4	C9		RET		

アドレス	機械語	ラベル	ニーモニック		備考
89D5	CD 56 8D	NGE-2	CALL	ADB	
8	E6 10		ANI	10	
A	C2 C8 89		JNZ	NGE-1	
D	7E		MOV	A,M	
E	E6 40		ANI	40	
E0	C2 C8 89		JNZ	NGE-1	
3	22 10 82		SHLD	8210	
6	C5		PUSH	B	
7	CD 4A 8A		CALL	AOT	
A	C1		POP	B	
B	A7		ANA	A	
C	C2 F7 89		JNZ	NGE-3	
F	EB		XCHG		
F0	22 10 82		SHLD	8210	
3	EB		XCHG		
4	C3 C8 89		JMP	NGE-1	
89F7	46	NGE-3	MOV	B,M	
8	36 11		MVI	M,11	
A	AF		XRA	A	
B	12		STAX	D	
C	3C		INR	A	
D	CD 43 8B		CALL	INR	
A00	C9		RET		
1	00 00 00		NOP	3	
4	00 00 00		NOP	3	
8A07	F5	OTE	PUSH	PSW	
8	D5		PUSH	D	
9	E5		PUSH	H	
A	57		MOV	D,A	
B	0E FF		MVI	C,FF	
D	5D		MOV	E,L	
8A0E	0C	OTE-1	INR	C	
8A0F	7A	OTE-2	MOV	A,D	
10	CD A0 8C		CALL	DRN	
3	FE FF		CPI	FF	
5	CA 37 8A		JZ	OTE-5	
8	4F		MOV	C,A	
9	CD 31 8D		CALL	ADA	
C	E6 1F		ANI	1F	
E	CA 0E 8A		JZ	OTE-1	
21	FE 11		CPI	11	
3	CA 2F 8A		JZ	OTE-3	
6	79		MOV	A,C	
7	E6 78		ANI	78	
9	C6 08		API	08	
B	4F		MOV	C,A	
C	C3 0F 8A		JMP	OTE-2	
8A2F	E1	OTE-3	POP	H	
30	D1		POP	D	
8A31	F1	OTE-4	POP	PSW	
2	77		MOV	M,A	
3	AF		XRA	A	
4	12		STAX	D	
5	47		MOY	B,A	
6	C9		RET		
8A37	E1	OTE-5	POP	H	
8	D1		POP	D	
9	CD 4A 8A		CALL	AOT	
C	A7		ANA	A	
D	CA 42 8A		JZ	OTE-6	
40	F1		POP	PSW	
1	C9		RET		
8A42	F1	OTE-6	POP	PSW	
3	77		MOV	M,A	
4	AF		XRA	A	
5	12		STAX	D	
6	C9		RET		
7	00 00 00		NOP	3	
8A4A	D5	AOT	PUSH	D	
B	1A		LDAX	D	
C	F5		PUSH	PSW	
D	E5		PUSH	H	
E	56		MOV	D,M	
F	77		MOV	M,A	
50	6B		MOV	L,E	
1	36 00		MVI	M,00	
3	2A 10 82		LHLD	8210	
6	5D		MOV	E,L	
7	0E 00		MVI	C,00	

アドレス	機械語	ラベル	ニーモニック		備考
9	C3 63 8A		JMP	AOT-2	
8A5C	0C	AOT-1	INR	C	
D	79		MOV	A,C	
E	FE 08		CPI	08	
60	CA 86 8A		JZ	AOT-4	
8A63	CD 56 8D	AOT-2	CALL	ADB	
6	E6 1F		ANI	1F	
8	CA 5C 8A		JZ	AOT-1	
B	FE 06		CPI	06	
D	CA D2 8A		JZ	AOT-8	
70	FE 02		CPI	02	
2	CA D2 8A		JZ	AOT-8	
5	FE 0A		CPI	0A	
7	CA D2 8A		JZ	AOT-8	
A	0E 08		MVI	C,08	
C	C3 86 8A		JMP	AOT-4	
8A7F	0C	AOT-3	INR	C	
80	79		MOV	A,C	
1	FE 20		CPI	20	
3	CA AD 8A		JZ	AOT-6	
8A86	CD 56 8D	AOT-4	CALL	ADB	
9	E6 1F		ANI	1F	
B	CA 7F 8A		JZ	AOT-3	
E	FE 02		CPI	02	
90	CA D2 8A		JZ	AOT-8	
3	FE 0A		CPI	0A	
5	CA D2 8A		JZ	AOT-8	
8	79		MOV	A,C	
9	E6 78		ANI	78	
B	C6 08		ADI	08	
D	4F		MOV	C,A	
E	FE 20		CPI	20	
A0	C2 86 8A		JNZ	AOT-4	
3	C3 AD 8A		JMP	AOT-6	
8AA6	0C	AOT-5	INR	C	
7	79		MOV	A,C	
8	FE 40		CPI	40	
A	CA CA SA		JZ	AOT-7	
8AAD	CD 56 8D	AOT-6	CALL	ADB	
B0	E6 1F		ANI	1F	
2	CA A6 8A		JZ	AOT-5	
5	FE 03		CPI	03	
7	CA D2 8A		JZ	AOT-8	
A	FE 0B		CPI	0B	
C	CA D2 8A		JZ	AOT-8	
F	79		MOV	A,C	
C0	E6 78		ANI	78	
2	C6 08		ADI	08	
4	4F		MOV	C,A	
5	FE 40		CPI	40	
7	C2 AD 8A		JNZ	AOT-6	
8ACA	E1	AOT-7	POP	H	
B	72		MOV	M,D	
C	F1		POP	PSW	
D	D1		POP	D	
E	12		STAX	D	
F	3E 01		MVI	A,01	
D1	C9		RET		
8AD2	45	AOT-8	MOV	B,L	
3	E1		POP	H	
4	72		MOV	M,D	
5	F1		POP	PSW	
6	D1		POP	D	
7	12		STAX	D	
8	AF		XRA	A	
9	C9		RET		
A	00 00 00		NOP	3	
8ADC	78	NFU	MOV	A,B	
E	E6 0F		ANI	0F	
E0	FE 07		CPI	07	
2	C2 FB 8A		JNZ	NFU-2	
5	C5		PUSH	B	
6	E5		PUSH	H	
7	0E 09		PUSH	C,09	
9	7D		MOV	A,L	
A	EB F0		ANI	F0	
C	6F		MOV	L,A	
8AED	7E	NFU-1	MOV	A,M	
E	E6 1F		ANI	1F	

アドレス	機械語	ラベル	ニーモニック		備考
F0	B8		CMP	B	
1	CA 04 8B		JZ	NFU-3	
4	2C		INR	L	
5	0D		DCR	C	
6	C2 ED 8A		JNZ	NFU-1	
9	E1		POP	H	
A	C1		POP	B	
8AFB	70	NFU-2	MOV	M,B	
C	5D		MOV	E,L	
D	68		MOV	L,B	
E	35		DCR	M	
F	CD 00 8C		CALL	KIK	
B02	37		STC		
3	C9		RET		
8B04	E1	NFU-3	POP	H	
5	C1		POP	B	
6	C9		RET		
7	00 00 00		NOP	3	
8B0A	1A	NRI	LDAX	D	
B	E6 08		ANI	08	
D	C2 23 8B		JNZ	NRI-1	
10	7D		MOV	A,L	
1	E6 0F		ANI	0F	
3	FE 02		CPI	02	
5	DA 32 8B		JC	NRI-3	
8	CA 25 8B		JZ	NRI-2	
B	7B		MOV	A,E	
C	E6 0F		ANI	0F	
E	FE 03		CPI	03	
20	DA 32 8B		JC	NRI-3	
8B23	AF	NRI-1	XRA	A	
4	C9		RET		
8B25	1A	NRI-2	LDAX	D	
6	E6 0F		ANI	0F	
8	FE 05		CPI	05	
A	CA 3D 8B		JZ	NRI-4	
D	FE 06		CPI	06	
F	CA 3D 8B		JZ	NRI-4	
8B32	1A	NRI-3	LDAX	D	
3	E6 0F		ANI	0F	
5	FE 04		CPI	04	
7	CA 3D 8B		JZ	NRI-4	
A	3E 01		MVI	A,01	
C	C9		RET		
8B3D	3E 02	NRI-4	MVI	A,02	
F	C9		RET		
40	00 00 00		NOP	3	
8B43	E5	INR	PUSH	H	
4	2E 7A		MVI	L,INR-1	
6	C3 4C 8B		JMP	DCR-1	
8B49	E5	DCR	PUSH	H	
A	2E 74		MVI	L,DCR-5	
8B4C	D5	DCR-1	PUSH	D	
D	F5		PUSH	PSW	
E	78		MOV	A,B	
F	E6 1F		ANI	1F	
51	CA 76 8B		JZ	DCR-6	
4	FE 10		CPI	10	
6	DA 65 8B		JC	DCR-2	
9	E6 0F		ANI	0F	
B	FE 08		CPI	08	
D	CA 6E 8B		JZ	DCR-4	
60	E6 07		ANI	07	
2	C3 6E 8B		JMP	DCR-4	
8B65	FE 08	DCR-2	CPI	08	
7	CA 6C 8B		JZ	DCR-3	
A	E6 07		ANI	07	
8B6C	F6 10	DCR-3	ORI	10	
8B6E	5F	DCR-4	MOV	E,A	
F	16 82		MVI	D,82	
71	26 8B		MVI	H,8B	
3	E9		PCHL		
8B74	EB	DCR-5	XCHG		
5	35		DCR	M	
8B76	F1	DCR-6	POP	PSW	
7	D1		POP	D	
8	E1		POP	H	
9	C9		RET		
8B7A	EB	INR-1	XCHG		

アドレス	機械語	ラベル	ニーモニック	備考
B	34		INR M	
C	F1		POP PSW	
D	D1		POP D	
E	E1		POP H	
F	C9		RET	
8B80 〜 8B97	空			
8	E5		PUSH H	
9	FE 01		CPI 01	
B	C2 A6 8B		JNZ SMB-1	
E	3D		DCR A	
F	12		STAX D	
A0	62		MOV H,D	
1	68		MOV L,B	
2	34		INR M	
3	3C		INR A	
4	E1		POP H	
5	C9		RET	
8BA6	26 82	SMB-1	MVI H,82	
8	6F		MOV L,A	
9	70		MOV M,B	
A	6B		MOV L,E	
B	72		MOV M,D	
C	54		MOV D,H	
D	CD 49 8B		CALL DCR	
B0	E1		POP H	
1	C9		RET	
2	00 00 00		NOP 3	
8BB5	00 00 00		NOP 3	
8BB8	E5	GKB	PUSH H	
9	FE 02		CPI 02	
B	DA C9 8B		JC GKB-1	
E	C2 D8 8B		JNZ GKB-2	
C1	62		MOV H,D	
2	6B		MOV L,E	
3	36 00		MVI M,00	
5	68		MOV L,B	
6	34		INR M	
7	E1		POP H	
8	C9		RET	
8BC9	2A 10 82	GKB-1	LHLD 8210	
C	70		MOV M,B	
D	6B		MOV L,E	
E	36 11		MVI M,11	
D0	22 10 82		SHLD 8210	
3	CD 49 8B		CALL DCR	
6	E1		POP H	
7	C9		RET	
8BD8	26 82	GKB-2	MVI H,82	
A	6F		MOV L,A	
B	72		MOV M,D	
C	6B		MOV L,E	
D	70		MOV M,B	
E	54		MOV D,H	
F	CD 49 8B		CALL DCR	
E2	E1		POP H	
3	C9		RET	
8BE4	00		NOP	
〜	〜		〜	
8BFF	00		NOP	
8C00	C5	KIK	PUSH B	
1	D5		PUSH D	
2	E5		PUSH H	
3	F5		PUSH PSW	
4	11 20 82		LXI D,8220	
7	06 09		MVI B,09	
8C09	0E 09	KIK-1	MVI C,09	
8C0B	1A	KIK-2	LDAX D	
C	E6 1F		ANI 1F	
E	12		STAX D	
F	1C		INR E	
10	0D		DCR C	
1	C2 0B 8C		JNZ KIK-2	
4	7B		MOV A,E	
5	C6 07		ADI 07	
7	5F		MOV E,A	
8	05		DCR B	
9	C2 09 8C		JNZ KIK-1	

アドレス	機械語	ラベル	ニーモニック	備考
C	1E 20		MVI E,20	
8C1E	06 09	KIK-3	MVI B,09	
8C20	0E 00	KIK-4	MVI C,00	
2	1A		LDAX D	
3	E6 1F		ANI 1F	
5	C2 3B 8C		JNZ KIK-6	
8C28	13	KIK-5	INX D	
9	05		DCR B	
A	C2 20 8C		JNZ KIK-4	
D	7B		MOV A,E	
E	C6 07		ADI 07	
30	5F		MOV E,A	
1	FE B0		CPI B0	
3	C2 1E 8C		JNZ KIK-3	
6	F1		POP PSW	
7	E1		POP H	
8	D1		POP D	
9	C1		POP B	
A	C9		RET	
8C3B	FE 11	KIK-6	CPI 11	
D	CA 28 8C		JZ KIK-5	
40	E6 10		ANI 10	
2	C2 76 8C		JNZ KIK-12	
8C45	1A	KIK-7	LDAX D	
6	CD A0 8C		CALL DRN	
9	FE FF		CPI FF	
B	CA 28 8C		JZ KIK-5	
E	4F		MOV C,A	
F	CD 31 8D		CALL ADA	
52	FE F9		CPI F9	
4	CA 69 8C		JZ KIK-10	
7	FE 40		CPI 40	
9	D2 61 8C		JNC KIK-8	
C	F6 40		ORI 40	
E	C3 63 8C		JMP KIK-9	
8C61	F6 80	KIK-8	ORI 80	
8C63	77	KIK-9	MOV M,A	
4	E6 0F		ANI 0F	
6	CA 72 8C		JZ KIK-11	
8C69	79	KIK-10	MOV A,C	
A	E6 78		ANI 78	
C	C6 08		ADI 08	
E	4F		MOV C,A	
F	C3 45 8C		JMP KIK-7	
8C72	0C	KIK-11	INR C	
3	C3 45 8C		JMP KIK-7	
8C76	1A	KIK-12	LDAX D	
7	CD A0 8C		CALL DRN	
A	FE FF		CPI FF	
C	CA 28 8C		JZ KIK-5	
F	4F		MOV C,A	
80	CD 56 8D		CALL ADB	
3	FE F9		CPI F9	
5	CA 90 8C		JZ KIK-13	
8	F6 20		ORI 20	
A	77		MOV M,A	
B	E6 0F		ANI 0F	
D	CA 99 8C		JZ KIK-14	
8C90	79	KIK-13	MOV A,C	
1	E6 78		ANI 78	
3	C6 08		ADI 08	
5	4F		MOV C,A	
6	C3 76 8C		JMP KIK-12	
8C99	0C	KIK-14	INR C	
A	C3 76 8C		JMP KIK-12	
D	00 00 00		NOP 3	
8CA0	C5	DRN	PUSH B	
1	E5		PUSH H	
2	E6 0F		ANI 0F	
4	06 08		MVI B,08	香
6	FE 06		CPI 06	
8	CA 1F 8D		JZ DRN-8	
B	06 20		MVI B,20	飛
D	FE 02		CPI 02	
F	CA 1F 8D		JZ DRN-8	
B2	FE 03		CPI 03	角
4	CA 19 8D		JZ DRN-7	
7	26 8C		MVI H,8C	
9	FE 0A		CPI 0A	

アドレス	機械語	ラベル	ニーモニック	備考
B	CA FC 8C		JZ DRN-3	
E	FE 0B		CPI 0B	成角
C0	CA 0B 8D		JZ DRN-5	
3	CD CF 8C		CALL DRN-2	
8CC6	7E	DRN-1	MOV A,M	
7	B9		CMP C	
8	D2 2B 8D		JNC DRN-10	
B	23		INX H	
C	C3 C6 8C		JMP DRN-1	
8CCF	2E E1	DRN-2	MVI L,DN-1	銀
D1	FE 04		CPI 04	
3	C8		RZ	
4	2E E7		MVI L,DN-2	桂
6	FE 05		CPI 05	
8	C8		RZ	
9	2E EA		MVI L,DN-3	歩
B	FE 07		CPI 07	
D	C8		RZ	
E	2E EC		MVI L,DN-4	金、成銀、桂、香、歩
E0	C9		RET	
E1	00 20 28	DN-1	00 20 28	
4	30 38 FF		30 38 FF	
E7	40 48 FF	DN-2	40 48 FF	
EA	00 FF	DN-3	00 FF	
EC	00 08 10	DN-4	00 08 10	
F	18 20 38		18 20 38	
F2	FF		FF	
F3	20 28 30	DN-5	20 28 30	
6	38 FF		38 FF	
F8	00 08 10	DN-6	00 08 10	
B	18		18	
8CFC	AF	DNR-3	XRA A	
8CFD	B9	DNR-4	CMP C	
E	D2 2B 8D		JNC DRN-10	(8DCB)
D01	3C		INR A	
2	05		DCR B	
3	C2 FD 8C		JNZ DRN-4	
6	2E F3		MVI L,DN-5	
8	C3 C6 8C		JMP DRN-1	
8D0B	2E F8	DNR-5	MVI L,DN-6	
D	06 04		MVI B,04	
8D0F	7E	DNR-6	MOV A,M	
10	B9		CMP C	
1	D2 2B 8D		JNC DRN-10	
4	23		INX H	
5	05		DCR B	
6	C2 0F 8D		JNZ DRN-6	
8D19	3E 20	DNR-7	MVI A,20	
B	47		MOV B,A	
C	C3 20 8D		JMP DRN-9	
8D1F	AF	DNR-8	XRA A	
8D20	B9	DNR-9	CMP C	
1	D2 2B 8D		JNC DRN-10	
4	3C		INR A	
5	05		DCR B	
6	C2 20 8D		JNZ DRN-9	
9	AF		XRA A	
A	3D		DCR A	
8D2B	E1	DRN-10	POP H	
C	C1		POP B	
D	C9		RET	
E	00 00 00		NOP 3	
8D31	C5	ADA	PUSH B	
2	79		MOV A,C	
3	E6 7F		ANI 7F	
5	FE 40		CPI 40	
7	CA 4A 8D		JZ ADA-1	
A	FE 48		CPI 48	
C	CA 50 8D		JZ ADA-2	
F	E6 07		ANI 07	
41	47		MOV B,A	
2	79		MOV A,C	
3	E6 78		ANI 78	
5	C6 B4		ADI B4	
7	C3 8F 8D		JMP ADD	
8D4A	7B	ADA-1	MOV A,E	
B	C6 0E		ADI 0E	
D	C3 96 8D		JMP ADD-1	
8D50	7B	ADA-2	MOV A,E	

アドレス	機械語	ラベル	ニーモニック	備考
1	D6 12		SUI 12	
3	C3 96 8D		JMP ADD-1	
8D56	C5	ADB	PUSH B	
7	79		MOV A,C	
8	E6 7F		ANI 7F	
A	FE 40		CPI 40	
C	CA 83 8D		JZ ADB-3	
F	FE 48		CPI 48	
61	CA 89 8D		JZ ADB-4	
4	E6 07		ANI 07	
6	47		MOV B,A	
7	79		MOV A,C	
8	E6 78		ANI 78	
A	FE 10		CPI 10	
C	DA 7E 8D		JC ADB-2	
F	FE 20		CPI 20	
71	DA 79 8D		JC ADB-1	
4	FE 30		CPI 30	
6	DA 7E 8D		JC ADB-2	
8D79	C6 A4	ADB-1	ADI A4	
B	C3 8F 8D		JMP ADD	
8D7E	C6 C4	ADB-2	ADI C4	
80	C3 8F 8D		JMP ADD	
8D83	7B	ADB-3	MOV A,E	
4	D6 0E		SUI 0E	
6	C3 96 8D		JMP ADD-1	
8D89	7B	ADB-4	MOV A,E	
A	C6 12		ADI 12	
C	C3 96 8D		JMP ADD-1	
8D8F	6F	ADD	MOV L,A	
90	26 8D		MVI H,8D	
2	7B		MOV A,E	
3	CD B3 8D		CALL ADD-3	
8D96	6F	ADD-1	MOV L,A	
7	26 82		MVI H,82	
9	FE 20		CPI 20	
B	DA AD 8D		JC ADD-2	
E	FE A9		CPI A9	
A0	D2 AD 8D		JNC ADD-2	
3	E6 0F		ANI 0F	
5	FE 09		CPI 09	
7	D2 AD 8D		JNC ADD-2	
A	7E		MOV A,M	
B	C1		POP B	
C	C9		RET	
8DAD	3E F9	ADD-2	MVI A,F9	
F	C1		POP B	
B0	C9		RET	
1	00		NOP	
2	00		NOP	
8DB3	E9	ADD-3	PCHL	
8DB4	3D	ADD-4	DCR A	
5	05		DCR B	
6	F2 B4 8D		JP ADD-4	
9	C9		RET	
A	00 00		NOP 2	
8DBC	D6 10	ADD-5	SUI 10	
E	05		DCR B	
F	F2 BC 8D		JP ADD-5	
C2	C9		RET	
3	00		NOP	
8DC4	3C	ADD-6	INR A	
5	05		DCR B	
6	F2 C4 8D		JP ADD-6	
9	C9		RET	
A	00 00		NOP 2	
8DCC	C6 10	ADD-7	ADI 10	
E	05		DCR B	
F	F2 CC 8D		JP ADD-7	
D2	C9		RET	
3	00		NOP	
8DD4	D6 11	ADD-8	SUI 11	
6	05		DCR B	
7	F2 D4 8D		JP ADD-8	
A	C9		RET	
B	00		NOP	
8DDC	D6 0F	ADD-9	SUI 0F	
E	05		DCR B	
F	F2 DC 8D		JP ADD-9	

アドレス	機械語	ラベル	ニーモニック	備考
E2	C9		RET	
3	00		NOP	
8DE4	C6 11	ADD-10	ADI 11	
6	05		DCR B	
7	F2 E4 8D		JP ADD-10	
A	C9		RET	
B	00		NOP	
8DEC	C6 0F	ADD-11	ADI 0F	
E	05		DCR B	
F	F2 EC 8D		JP ADD-11	
F2	C9		RET	
3	00		NOP	
8DF4	00		NOP	
⟆	⟆		⟆	
8DF8	00		NOP	
8DF9	02 02 04	MGM	02 02 04	
C	04 04 12		04 04 12	
F	04		04	
8E00	21 00 82	CLR	LXI H,8200	
3	06 B0		MVI B,B0	
5	AF		XRA A	
8E06	77	CLR-1	MOV M,A	
7	23		INX H	
8	05		DCR B	
9	C2 06 8E		JNZ CLR-1	
C	FF		RST 7	
D	00 00		NOP 2	
8E0F	00 00 00		NOP 3	
12	21 F9 8D	SET	LXI H,MGM	
5	11 12 82		LXI D,8212	
8	06 07		MVI B,07	
8E1A	7E	SET-1	MOV A,M	
B	12		STAX D	
C	13		INX D	
D	23		INX H	
E	05		DCR B	
F	C2 1A 8E		JNZ SET-1	
22	1B		DCX D	
3	21 08 82		LXI H,8208	
6	0E 07		MVI C,07	
8E28	1A	SET-2	LDAX D	
9	96		SUB M	
A	12		STAX D	
B	1B		DCX D	
C	2B		DCX H	
D	0D		DCR C	
E	C2 28 8E		JNC SET-2	
31	2E 20		MVI L,20	
3	16 09		MVI D,09	
8E35	1E 09	SET-3	MVI E,09	
8E37	7E	SET-4	MOV A,M	
8	E6 0F		ANI 0F	
A	CA 4D 8E		JZ SET-6	
D	FE 01		CPI 01	
F	CA 49 8E		JZ SET-5	
42	47		MOV B,A	
3	CD 49 8B		CALL DCR	
6	C3 4D 8E		JMP SET-6	
8E49	22 10 82	SET-5	SHLD 8210	
C	0C		INR C	
8E4D	23	SET-6	INX H	
E	1D		DCR E	
F	C2 37 8E		JNZ SET-4	
52	7D		MOV A,L	
3	C6 07		ADI 07	
5	6F		MOV L,A	
6	15		DCR D	
7	C2 35 8E		JNZ SET-3	
A	0D		DCR C	
8E5B	C8		RZ	
C	FF		RST 7	
D	00 00		NOP 2	これ以降前のプログラムと同じ部分はありません。
F	00 00		NOP 2	
8E61	75 75 20			
4	4B 7C 6C			
7	76 78 20			
A	77 5E 5D			
D	79 72 4F			
70	77 6E 73			

アドレス	機械語	ラベル	ニーモニック	備考
3	4C 20 20			
6	77 5D 20			
9	20 20 20			
C	58 6D 73			
F	73 4F 20			
8E82	FE 01	SMD	CPI 01	
4	C2 8B 8E		JNZ SMD-1	
7	CD AD 8E		CALL UCH	
A	C9		RET	
8E8B	47	SMD-1	MOV B,A	
C	7B		MOV A,E	
D	58		MOV E,B	
E	CD BF 8E		CALL IDO	
91	C9		RET	
8E92	FE 02	GKD	CPI 02	
4	CA A7 8E		JZ GKD-2	
7	D2 A3 8E		JNC GKD-1	
A	E5		PUSH H	
B	CD 56 8D		CALL ADB	
E	7B		MOV A,E	
F	5D		MOV E,L	
A0	16 01		MVI D,01	
2	E1		POP H	
8EA3	CD BF 8E	GKD-1	CALL IDO	
6	C9		RET	
8EA7	CD AD 8E	GKD-2	CALL UCH	
A	C9		RET	
B	00 00		NOP 2	
8EAD	23	UCH	INX H	
E	23		INX H	
F	23		INX H	
B0	23		INX H	
1	7B		MOV A,E	
2	CD DB 8E		CALL DCA	
5	78		MOV A,B	
6	CD F4 8E		CALL DCB	
9	36 73		MVI M,73	
B	23		INX H	
C	36 41		MVI M,41	
E	C9		RET	
8EBF	CD DB 8E	IDO	CALL DCA	
C2	7B		MOV A,E	
3	CD DB 8E		CALL DCA	
6	7A		MOV A,D	
7	E6 0F		ANI 0F	
9	FE 0C		CPI 0C	
B	D4 0C 8F		CNC DCC	
E	CD F4 8E		CALL DCB	
D1	79		MOV A,C	
2	E6 80		ANI 80	
4	C4 0C 8F		CNZ DCC	
7	C9		RET	
8	00 00 00		NOP 3	
8EDB	C5	DCA	PUSH B	
C	47		MOV B,A	
D	E6 F0		ANI F0	
F	0F		RRC	
E0	0F		RRC	
1	0F		RRC	
2	0F		RRC	
3	C6 2F		ADI 2F	
5	77		MOV M,A	
6	23		INX H	
7	36 2D		MVI M,2D	
9	23		INX H	
A	78		MOV A,B	
B	E6 0F		ANI 0F	
D	C6 E0		ADI E0	
F	77		MOV M,A	
F0	23		INX H	
1	23		INX H	
2	C1		POP B	
3	C9		RET	
8EF4	D5	DCB	PUSH D	
5	E6 0F		ANI 0F	
7	57		MOV D,A	
8	82		ADD D	
9	82		ADD D	
A	C6 5E		ADI 5E	

アドレス	機 械 語	ラベル	ニーモニック	備 考
C	5F		MOV E,A	
D	16 8E		MVI D,8E	
F	06 03		MVI B,03	
8F01	1A	DCB-1	LDAX D	
2	77		MOV M,A	
3	13		INX D	
4	23		INX H	
5	05		DCR B	
8F06	C2 01 8F		JNZ DCB-1	
9	23		INX H	
A	D1		POP D	
B	C9		RET	
8F0C	E6 07	DCC	ANI 07	
E	36 45		MVI M,45	
10	23		INX H	
1	36 58		MVI M,58	
3	23		INX H	
4	C9		RET	
8F15	3E 06	DPA	MVI A,06	
7	32 7A 84		STA 847A	
A	21 24 8F		LXI H,TMN	
D	22 7B 84		SHLD 847B	
20	CD 52 FA		CALL FA52	
3	C9		RET	
8F24	C2 D0 C5	TMN		
7	BC 0D 0D			
8F2A	3E 0B	DPB	MVI A,0B	
C	CD 04 F8		CALL F804	
F	C9		RET	
8F30	AF	SMI	XRA A	
1	4F		MOV C,A	
2	3C		INR A	
3	06 02		MVI B,02	
5	16 82		MVI D,82	
7	C9		RET	
8	00 00 00		NOP 3	
8F3B	FE 01	SMC	CPI 01	
D	C2 42 8F		JNZ SMC-1	
40	0C		INR C	
1	C9		RET	
8F42	E5	SMC-1	PUSH H	
3	6F		MOV L,A	
4	79		MOV A,C	
5	FE 80		CPI 80	
7	D2 59 8F		JNC SMC-2	
A	62		MOV H,D	
B	CD 0A 8B		CALL NRI	
E	A7		ANA A	
F	CA 5C 8F		JN SMC-3	
52	79		MOV A,C	
3	F6 80		ORI 80	
5	4F		MOV C,A	
6	7D		MOV A,L	
7	E1		POP H	
8	C9		RET	
8F59	E6 7F	SMC-2	ANI 7F	
B	4F		MOV C,A	
8F5C	78	SMC-3	MOV A,B	
D	E6 0F		ANI 0F	
F	CA 6B 8F		JZ SMC-4	
62	79		MOV A,C	
3	E6 78		ANI 78	
5	C6 08		ADI 08	
7	4F		MOV C,A	
8	AF		XRA A	
9	E1		POP H	
A	C9		RET	
8F6B	0C	SMC-4	INR C	
C	AF		XRA A	
D	E1		POP H	
E	C9		RET	
F	00 00 00		NOP 3	
8F72	16 82	GKI	MVI D,82	
4	FE 01		CPI 01	
6	CA 8D 8F		JZ GKI-2	
9	5F		MOV E,A	
A	4C		MOV C,H	
B	7D		MOV A,L	
C	A7		ANA A	
D	C2 8C 8F		JNZ GKI-1	
80	2A 10 82		LHLD 8210	
3	7E		MOV A,M	
4	E6 80		ANI 80	
6	C2 A9 8F		JNZ GKI-5	
9	C3 8D 8F		JMP GKI-2	
8F8C	5F	GKI-1	MOV E,A	
8F8D	1A	GKI-2	LDAX D	
E	47		MOV B,A	
F	E6 20		ANI 20	
91	CA 97 8F		JZ GKI-3	
4	AF		XRA A	
5	4F		MOV C,A	
6	C9		RET	
8F97	79	GKI-3	MOV A,C	
8	E6 07		ANI 07	
A	CA A6 8F		JZ GKI-4	
D	79		MOV A,C	
E	E6 78		ANI 78	
A0	4F		MOV C,A	
1	06 12		MVI B,12	
3	3E 02		MVI A,02	
5	C9		RET	
8FA6	2A 10 82	GKI-4	LHLD 8210	
8FA9	5D	GKI-5	MOV E,L	
A	AF		XRA A	
B	4F		MOV C,A	
C	3C		INR A	
D	C9		RET	
E	00 00 00		NOP 3	
8FB1	FE 01	GKC	CPI 01	
3	CA CB 8F		JZ GKC-2	
6	FE 02		CPI 02	
8	CA C6 8F		JZ GKC-1	
B	79		MOV A,C	
C	E6 78		ANI 78	
E	C6 08		ADI 08	
C0	4F		MOV C,A	
1	AF		XRA A	
2	CD 00 8C		CALL KIK	
5	C9		RET	
8FC6	0C	GKC-1	INR C	
7	CD 00 8C		CALL KIK	
A	C9		RET	
8FCB	79	GKC-2	MOV A,C	
C	C6 08		ADI 08	
E	4F		MOV C,A	
F	3E 01		MVI A,01	
D1	CD 00 8C		CALL KIK	
4	C9		RET	
5	00 00 00		NOP 3	
8FD8			以上:2520バイト	

32

数値による知的ゲーム

スタートレック・ゲーム

TK-80BS＋M20K

鍵富靖雄

TK-80を購入した時より，いずれスタートレックを動かしたいと思っておりましたが，その後TK-80BS，TKM-20Kと増設し，ハードとしてはメモリ量も充分となり，スタートレックが動かせる様になりました。そこで念願のスタートレックに挑戦というわけです。

スタートレックのプログラムはパロ・アルト版ベーシックの8Kスタートレック及びアルティア版ベーシック・スタートレックのリストが発表されております。

今回はアルティア版をNECレベルIIベーシックに乗る様に一部を書き換えました。なお，このプログラムはプログラムで9.3Kバイト，配列で約1Kバイト，計11Kバイトは必要と思われます。その他，ベーシックのワーキングエリアとして1Kバイト必要ですので，RAMは総計12Kは用意して下さい。

―――アルティア版をNECレベルII版に移植する時の注意―――

① NECレベルIIには関数指定，FND(　)命令はないので，関数が必要な時はそのつどプログラムする。

② 130行はオリジナルでは，K=K+ (N<×2)+ (N<Y2)+ (N<. 28)+ (N<. 08)+ (N<. 03)+(N<. 01)：K9=K9−K：GOTO 160 となっています。(N<×2)，(N<. 28) 等の意味は（　）の中が成立した時（　）は−1，成立しない時は0ということです。そこで1500行へとばしてIF　T. 文でF1～F6を−1又は0にして132行へもどり，それらの加算をさせています。

なおこのルーチンはクリンゴンの発生確率を定めています。

③ 1340～1344行，及び1580行以上ですが，オリジナルでは　E$=STR$ (Q, (I, J)：E$="00"+MID$ (E$, 2)：PRINT RIGHT$(E$, 3)；となっております。TK-80BSのベーシックには残念ながらE$=STR$ (……)の命令はありませんので，Q (I, J)=0の時は000を書きQ (I, J)<10の時は00を書いた後，1～9の文字列(20行で指定してあるX$)よりQ (I, J) 番目の文字を書かせます。Q (I, J)<100の時は0を書いた後Q(I, J)／10の整数と，Q (I, J) −Q (I, J)／10の整数を作り，X$の文字例の中からそれぞれ選んで書かせます。Q(I, J)≧100のときも同様の考えで行っています。なおQ (I, J) とはコードラント内のクリンゴンの数，ベースの数，星の数を示す3桁以下の数字でQ (I, J)=213とはクリンゴン　2，ベース　1，星　3を意味します。なぜこの様な苦労をするかといいますと，＊＊＊表示とQ (I, J) の表示を画面上で位置をうまく合わせる為とQ (I, J)=1の時は001と表示させる為です。

④ プリント文の前後にPOKE 8624H, 00Hがところどころに入っていますが，レベルIIでは

P. "……32文字以上……"；"……"

を実行させますと，前の文字グループの後に；が入っているにもかかわらず，前の文字グループと後の文字グループが連続して表示されません，それを防ぐため8624番地のカウンターをクリアしています。それから380行あたりでのPRINT "DATE="；T；"　"の最後の"　"は一見意味が無い様ですが，これも必要です，実験して見て下さい。

⑤ レベルIIでは－のべき乗は出来ません（－2)^2 はダメ。

その他細かい点で変えてある所がかなりあります。

———スタートレックの遊び方——

我々は今エンタープライズの艦長です。囲りは小宇宙（SECTOR)，と大宇宙（QUADRANT）**第32—1図**です。小宇宙，大宇宙とも8×8の座標で構成されて大宇宙の各々の一つが小宇宙です。又宇宙の外は無の世界です。本スタートレックをプリンター上で行った結果を見ながら練習航海へ出ましょう。

まずWORKINGでしばらくお休みです。

命令（OBJECTIVE）が出ました。"48光年で48機のクリンゴンをせん滅せよ""基地は4"，続いて現在の小宇宙（SECTOR）の状況を出力してきました。星（※）が5でクリンゴン（K）はいません。

長距離レーダーで隣接コードラントを策敵します(コマンド 3)，長距離レーダーの表示で1桁目はクリンゴンの数，2桁目はベースの数，3桁目は星の数です。現在のコードラント（1—3）は中央です。直下のコードラント（2—3）は305を表示，クリンゴン3機発見，2--3へ移動（WARP）（コマンド 1)コースを指定してやります。コースは**第32—1図**で示す様に決められています。直下ですから7です。移動量（WARP）を聞いて来ました。移動量は直線で1コードラント航行するに1ワープ必要です。直線でない時はピタゴラスの定理で計算します。今回は1—3から2—3で直線ですから1ワープでOKです。2—3のコードラントへ突入です。クリンゴンが巧撃して来ます。エンタープライズのエネルギーが減少します。我々も光子魚雷で巧撃しましょう。(コマンド 5) コースを指定します。コースは移動のときと同様第1回で定められています。この方向にクリンゴンがいるので2を入力します。魚雷は7—2，7—2，6—3，5—4，4—5と進み4—5のクリンゴンを撃墜しました。又クリンゴンの巧撃を受けました。我々も魚雷2発を発射して残りのクリンゴンをせん滅できました。コマン

《第32—1》各コマンド

ド表を出力してみます。7以上の数を入力するとコマンド表が表示されます(0をいれるとストップします，これはデバッグのため)。

長距離レーダーで策敵（コマンド　3)，コードラント2—2へ移動（コマンド　1)，光子魚雷発射（コマンド　5）1機撃墜，フェーザー砲発射準備（コマンド　4)エネルギー1000にセット発射，撃墜。

全銀河系宇宙を表示（コマンド　6)。今まで長距離レーダーで探査したコードラントの状況が表示されます。＊＊＊は未だ不明です。

この様にして，ゲームはドラマチックに進行します。宇宙嵐で各部が故障することもあります。エネルギーと魚雷は基地の隣へ行けば補給を受けられます。1回移動するとワープには関係なく1光年減少します。命令の年限内で，かつエネルギーで，アドミラルになれます。キャビンボーイに降格されない様がんばって下さい。

以上NECレベルII版スタートレックのリスト及び概要を書きましたが，プログラムの詳細について研究されたい方は参考文献をごらんになって下さい。

注意！！　プログラムリストで※→＊，￥→$，と読み直して下さい。それからIF文の後の>,<は),(になっています。私のプリンター（NEC E301）にはこれらの活字が無いのです。

参考文献

1）石田晴久　マイクロコンピュータのプログラミング　共立出版

2）池孝三　マイコンを楽しむためのベーシックマスター　学習研究社

スタートレック・プログラムリスト

```
    1 REM
    2 REM
    3 REM
   10 REM .****STERTREC****
   20 REM . A GAME OF INTRAGALACTIC WARF
ARE BASED ON NBC,S POPULAR TV CHOW.
   30 REM . ADAPTED FOR NEC LEVEL 2 BASI
C BY Y.K
   40 RANDOMIZE
   50 DIM D(5),L(7),M(7),N(7),S(7,7),Q(7
,7),D$(5)
   60 CLEAR
   70 LET Q$=".EKB*": LET X$="123456789"

   80 LET D$(0)="WARP ENGINS"
   90 LET D$(1)="SHORT RANGE SENSOR"
  100 LET D$(2)="LONG RANGE SENSORS"
  110 LET D$(3)="PHASERS"
  120 LET D$(4)="TORPEDOS": LET D$(5)="G
ALACTIC RECORDS"
  130 LET I=INT(RND(90))
  140 PRINT "  ****STERTREC****"
  150 PRINT
  160 PRINT
  170 LET I=I-11*INT(I/11): FOR J=0 TO I
: LET K=RND(1): NEXT J: PRINT "WORKING"
  180 GOSUB 1080
  190 GOSUB 860
  200 LET Q1=X: LET Q2=Y: LET X=8: LET Y
=1: LET X1=.2075: LET Y1=6.28: LET X2=3.
28
  210 LET Y2=1.8: LET A=.96: LET C=100:
LET W=10: LET K9=0: LET B9=0: LET S9=400
: LET T9=3451: GOTO 360
  220 IF F<X2 THEN  LET F1=-1
  230 IF F>=X2 THEN  LET F1=0
  240 IF F<Y2 THEN  LET F2=-1
  250 IF F>=Y2 THEN  LET F2=0
  260 IF F<.28 THEN  LET F3=-1
  270 IF F>=.28 THEN  LET F3=0
  280 IF F<.08 THEN  LET F4=-1
  290 IF F>=.08 THEN  LET F4=0
  300 IF F<.03 THEN  LET F5=-1
  310 IF F>=.03 THEN  LET F5=0
  320 IF F<.01 THEN  LET F6=-1
  330 IF F>=.01 THEN  LET F6=0
  340 LET K=K+F1+F2+F3+F4+F5+F6
  350 LET K9=K9-K: GOTO 420
  360 LET T0=3421: LET T=T0: LET E0=4000
: LET E=E0: LET P0=10: LET P=P0
  370 FOR I=0 TO 7
  380 FOR J=0 TO 7: LET K=0: LET F=RND(Y
)
  390 IF F<Y1 THEN  LET F1=-1
  400 IF F>=Y1 THEN  LET F1=0
  410 IF F<X1 THEN  LET F=F*64: LET K=F1
-Y: GOTO 220
  420 IF RND(Y)>A THEN  LET F1=-1
  430 IF RND(Y)<=A THEN  LET F1=0
  440 LET B=F1: LET B9=B9-B
  450 LET Q(I,J)=K*C+B*W-INT(RND(Y)*X+Y)
: NEXT J: NEXT I
  460 IF K9>(T9-T0) THEN  LET T9=T0+K9
  470 IF B9>0 THEN 500
  480 GOSUB 860
  490 LET Q(X,Y)=Q(X,Y)-10: LET B9=1
  500 LET K0=K9
  510 PRINT "OBJECTIVE: DESTROY";K9;
  520 PRINT "KLINGON BATTLE CRUSERS IN";
T9-T0;
  530 PRINT "YEARS,": PRINT "THE NUMBER
OF STARBASES IS";B9
  540 LET A=0
  550 IF Q1<0 THEN  LET F=0: LET S=0: LE
T K=0: GOTO 610
  560 IF Q1>7 THEN  LET F=0: LET S=0: LE
T K=0: GOTO 610
  570 IF Q2<0 THEN  LET F=0: LET S=0: LE
T K=0: GOTO 610
  580 IF Q2>7 THEN  LET F=0: LET S=0: LE
```

```
T K=0: GOTO 610
 590 LET F=ABS(Q(Q1,Q2)): LET Q(Q1,Q2)=
F
 600 LET S=F-INT(F/10)*10: LET K=INT(F/
100)
 610 LET B=INT(F/10-K*10): GOSUB 860
 620 LET S1=X: LET S2=Y
 630 FOR I=0 TO 7: FOR J=0 TO 7
 640 LET S(I,J)=1: NEXT J: NEXT I
 650 LET S(S1,S2)=2
 660 FOR I=0 TO 7: LET N(I)=0: LET X=8
 670 IF I<K THEN  GOSUB 870: LET S(X,Y)
=3: LET N(I)=S9
 680 LET L(I)=X: LET M(I)=Y: NEXT I
 690 LET I=S
 700 IF B>0 THEN  GOSUB 870: LET S(X,Y)
=4
 710 IF I>0 THEN  GOSUB 870: LET S(X,Y)
=5: LET I=I-1: GOTO 710
 720 GOSUB 990: IF A=0 THEN  GOSUB 890
 730 IF E<=0 THEN 2120
 740 LET I=1: IF D(I)>0 THEN 1090
 750 FOR I=0 TO 7: FOR J=0 TO 7: PRINT
MID(Q$,S(I,J),1);" ";: NEXT J
 760 PRINT " ";: ON I+1 GOTO 770,790,80
0,810,820,830,840,850
 770 PRINT "YEARS=";T9-T;" "
 780 LET F3=F3+1: NEXT I: GOTO 1120
 790 PRINT "DATE=";T;" ": GOTO 780
 800 PRINT "COND: ";C$: GOTO 780
 810 PRINT "QUAD=";Q1+1;Q2+1;" ": GOTO
780
 820 PRINT "SECT=";S1+1;S2+1;" ": GOTO
780
 830 PRINT "ENGY=";INT(E);" ": GOTO 780

 840 PRINT D$(4);"=";P;" ": GOTO 780
 850 PRINT "KLINGONS=";K9;" ": GOTO 780

 860 LET X=INT(RND(1)*8): LET Y=INT(RND
(1)*8): RETURN
 870 GOSUB 860: IF S(X,Y)>1 THEN 870
 880 RETURN
 890 IF K<1 THEN  RETURN
 900 IF C$="DOCED" THEN  PRINT "STARBAS
E PROTECTS ENTERPRISE": RETURN
 910 FOR I=1 TO 7: IF N(I)<=0 THEN  NEX
T I: RETURN
 920 LET H=N(I)*.4*RND(1): LET N(I)=N(I
)-H
 930 LET H1=SQR((L(I)-S1)*(L(I)-S1)+(M(
I)-S2)*(M(I)-S2))
 940 LET H=H/(H1^.4): LET E=E-H
 950 LET E$="ENTERORISE FROM": LET F=E:
 GOSUB 960: NEXT I: RETURN
 960 PRINT H;"UNIT HIT ON ";: POKE 8624
H,00H: PRINT E$;" SECTOR";L(I)+1;"-";M(I
)+1;
 970 POKE 8624H,00H
 980 PRINT "(";F;")LEFT": RETURN
 990 FOR I=S1-1 TO S1+1: FOR J=S2-1 TO
S2+1
 1000 IF I<0 THEN 1050
 1010 IF I>7 THEN 1050
 1020 IF J<0 THEN 1050
 1030 IF J>7 THEN 1050
 1040 IF S(I,J)=4 THEN  LET C$="DOCKD":
 LET E=E0: LET P=P0: GOSUB 1080: RETURN
 1050 NEXT J: NEXT I: IF K>0 THEN  LET C
$="RED": RETURN
 1060 IF E<E0*.1 THEN  LET C$="YELLOW":
RETURN
 1070 LET C$="GREEN": RETURN
 1080 FOR F=0 TO 5: LET D(F)=0: NEXT F:
RETURN
 1090 PRINT D$(I);" DAMAGED,";
 1100 PRINT " ";D(I);"YEARS ESTIMATED FO
R REPAIR.": PRINT
 1110 IF A=1 THEN  RETURN
 1120 INPUT "COMAND"A: PRINT
 1130 IF A=0 THEN  STOP
 1140 IF A<1 THEN 1170
 1150 IF A>6 THEN 1170
 1160 ON A GOTO 1200,720,1910,1780,1180,
1990
 1170 FOR I=0 TO 5: PRINT I+1;"=";D$(I):
 NEXT I: GOTO 1120
 1180 IF D(4)>0 THEN  PRINT "SPACE CRUD
BLOCKING TUBES.";: LET I=4: GOTO 1100
 1190 LET F=15: IF P<1 THEN  PRINT "NO T
ORPEDOES LEFT": GOTO 1120
 1200 IF A=5 THEN  PRINT "TORPEDO ";
 1210 INPUT "COURSE(1-8.9)"C: IF C<1 THE
N 1120
 1220 IF C>=9 THEN 1200
 1230 IF A=5 THEN  LET P=P-1: PRINT "TRA
CK:";: GOTO 1430
 1240 INPUT "WARP(0-12)"W
 1250 PRINT
 1260 IF W=0 THEN 1240
 1270 IF W>12 THEN 1200
 1280 IF W<=.2 THEN 1310
 1290 IF D(0)<=0 THEN 1310
 1300 LET I=0: PRINT D$(I);" DAMAGED, MA
X IS 0.2";: GOSUB 1100: GOTO 1240
 1310 GOSUB 890: IF E<=0 THEN 2120
 1320 IF RND(1)>.25 THEN 1400
 1330 LET X=INT(RND(1)*6): IF RND(1)>.5
THEN 1360
 1340 LET D(X)=D(X)+INT(6-RND(1)*5): PRI
NT "**SPACE STORM! ";
 1350 PRINT D$(X);: POKE 8624H,00H: PRIN
T " DAMAGED**": LET I=X: GOSUB 1100: LET
 D(X)=D(X)+1: GOTO 1400
 1360 FOR I=X TO 5: IF D(I)>0 THEN 1390
 1370 NEXT I
 1380 FOR I=0 TO X: IF D(I)<=0 THEN  NEX
T I: GOTO 1400
 1390 LET D(I)=.5: PRINT "SPOCK USED A N
EW REPAIR TECHNIQUE**"
 1400 FOR I=0 TO 5: IF D(I)=0 THEN 1420
 1410 LET D(I)=D(I)-1: IF D(I)<=0 THEN
LET D(I)=0: PRINT D$(I);" ARE FIXED!"
 1420 NEXT I: LET F=INT(W*8): LET E=E-F-
F+.5: LET T=T+1: LET S(S1,S2)=1
 1430 LET Y1=S1+.5: LET X1=S2+.5: IF T>T
9 THEN 2120
 1440 LET Y=(C-1)*.785396: LET X=COS(Y):
 LET Y=-SIN(Y)
 1450 FOR I=1 TO F: LET Y1=Y1+Y: LET X1=
X1+X: LET Y2=INT(Y1): LET X2=INT(X1)
 1460 IF X2<0 THEN 1670
 1470 IF X2>7 THEN 1670
 1480 IF Y2<0 THEN 1670
 1490 IF Y2>7 THEN 1670
 1500 IF A=5 THEN  PRINT Y2+1;"-";X2+1;
```

```
 1510 IF S(Y2,X2)=1 THEN  NEXT I: GOTO 1
620
 1520 PRINT " ": IF A=1 THEN  PRINT "BLO
CKED BY";
 1530 ON S(Y2,X2)-2 GOTO 1540,1600,1580
 1540 PRINT "KLINGON";: IF A=1 THEN 1610

 1550 FOR I=0 TO 7: IF Y2<>L(I) THEN 157
0
 1560 IF X2=M(I) THEN  LET N(I)=0
 1570 NEXT I: LET K=K-1: LET K9=K9-1: GO
TO 1630
 1580 PRINT "STAR";: IF A=5 THEN  LET S=
S-1: GOTO 1630
 1590 GOTO 1610
 1600 PRINT "STARBASE";: IF A=5 THEN  LE
T B=2: GOTO 1630
 1610 PRINT " AT SECTOR";Y2+1;"-";X2+1:
LET Y2=INT(Y1-Y): LET X2=INT(X1-X)
 1620 LET S1=Y2: LET S2=X2: LET S(S1,S2)
=2: LET A=2: GOTO 720
 1630 PRINT " DESTROYED!";: IF B=2 THEN
 LET B=0: PRINT "   GOOD WORK!";
 1640 PRINT " ": LET S(Y2,X2)=1: LET Q(Q
1,Q2)=K*100+B*10+S: IF K9<1 THEN 2190
 1650 GOSUB 890: IF E<=0 THEN 2120
 1660 GOSUB 990: GOTO 1120
 1670 IF A=5 THEN  PRINT "MISSED!": GOTO
 1650
 1680 LET Q1=INT(Q1+W*Y+(S1+.5)/8): LET
Q2=INT(Q2+W*X+(S2+.5)/8)
 1690 IF Q1<0 THEN  LET F1=-1
 1700 IF Q1>=0 THEN  LET F1=0
 1710 IF Q1>7 THEN  LET F2=-1
 1720 IF Q1<=7 THEN  LET F2=0
 1730 IF Q2<0 THEN  LET F3=-1
 1740 IF Q2>=0 THEN  LET F3=0
 1750 IF Q2>7 THEN  LET F4=-1
 1760 IF Q2<=7 THEN  LET F4=0
 1770 LET Q1=Q1-F1+F2: LET Q2=Q2-F3+F4:
GOTO 540
 1780 LET I=3: IF D(I)>0 THEN 1090
 1790 INPUT "PHASERS READY! ENERGY UNITS
 TO FIRE"X: IF X<=0 THEN 1120
 1800 PRINT
 1810 IF X>E THEN  PRINT "ONLY GOT";E: G
OTO 1790
 1820 LET E=E-X: LET Y=K: FOR I=0 TO 7:
IF N(I)<=0 THEN 1890
 1830 LET F3=(L(I)-S1)*(L(I)-S1): LET F4
=(M(I)-S2)*(M(I)-S2): LET F1=SQR(F3+F4)
 1840 LET H=X/(Y*(F1^.4)): LET N(I)=N(I)
-H
 1850 LET E$="KLINGON AT": LET F=N(I): G
OSUB 960
 1860 IF N(I)>0 THEN 1890
 1870 PRINT "**KLINGON DESTOROYED**"
 1880 LET K=K-1: LET K9=K9-1: LET S(L(I)
,M(I))=1: LET Q(Q1,Q2)=Q(Q1,Q2)-100
 1890 NEXT I: IF K9<1 THEN 2190
 1900 GOTO 1650
 1910 LET I=2: IF D(I)>0 THEN 1090
 1920 PRINT D$(I);" FOR QUADRANT";Q1+1;"
-";Q2+1
 1930 FOR I=Q1-1 TO Q1+1: FOR J=Q2-1 TO
Q2+1: PRINT "   ";
 1940 IF I<0 THEN  PRINT "***";: GOTO 20
80
 1950 IF I>7 THEN  PRINT "***";: GOTO 20
80
 1960 IF J<0 THEN  PRINT "***";: GOTO 20
80
 1970 IF J>7 THEN  PRINT "***";: GOTO 20
80
 1980 LET Q(I,J)=ABS(Q(I,J)): GOTO 2030
 1990 LET I=5: IF D(I)>0 THEN 1090
 2000 PRINT "CUMULATIVE GALACTIC MAP FOR
 STARDATE";: POKE 8624H,00H: PRINT T
 2010 FOR I=0 TO 7: LET H1=PEEK(847DH):
LET H1=H1-1: POKE 847DH,H1: FOR J=0 TO 7
: PRINT " ";
 2020 IF Q(I,J)<0 THEN  PRINT "***";: GO
TO 2080
 2030 IF Q(I,J)=0 THEN  PRINT "000";: GO
TO 2080
 2040 IF Q(I,J)<10 THEN  PRINT "00";: PR
INT MID(X$,Q(I,J),1);: GOTO 2080
 2050 IF Q(I,J)<100 THEN 2260
 2060 GOTO 2290
 2070 PRINT MID(X$,H1,1);: PRINT MID(X$,
H2,1);: PRINT MID(X$,H3,1);
 2080 NEXT J: PRINT
 2090 NEXT I: GOTO 1120
 2100 PRINT
 2110 PRINT "IT IS STARDATE ";T: RETURN

 2120 GOSUB 2100: PRINT "THANKS TO YOUR
BUNGLING. THE FEDERATION WILL BE";
 2130 POKE 8624H,00H
 2140 PRINT " ";
 2150 PRINT "CONQUERED BY THE REMAINING"
;K9;
 2160 POKE 8624H,00H
 2170 PRINT "KLINGONS CRUISERS"
 2180 PRINT "YOU ARE DEMOTED TO CABIN BO
Y!": GOTO 2230
 2190 GOSUB 2100: PRINT "THE FEDERATION
HAS BEEN SAVED!"
 2200 PRINT "YOU ARE PROMOTED TO ADMIRAL
": PRINT K0;"KLINGONS IN";
 2210 POKE 8624H,00H
 2220 PRINT T-T0;"YEARS. RATING=";INT(K0
/(T-T0)*1000)
 2230 INPUT "TRY AGAIN"E$
 2240 IF E$="YES" THEN 180
 2250 IF E$<>"YES" THEN  STOP
 2260 LET H1=INT(Q(I,J)/10): LET H2=Q(I,
J)-H1*10: PRINT "0";
 2270 IF H2=0 THEN  PRINT MID(X$,H1,1);:
 PRINT "0";: GOTO 2080
 2280 PRINT MID(X$,H1,1);: PRINT MID(X$,
H2,1);: GOTO 2080
 2290 LET H1=INT(Q(I,J)/100): LET H2=INT
((Q(I,J)-H1*100)/10): LET H3=Q(I,J)-H1*1
00-H2*10
 2300 IF Q(I,J)-H1*100=0 THEN  PRINT MID
(X$,H1,1);: PRINT "00";: GOTO 2080
 2310 IF H2=0 THEN  PRINT MID(X$,H1,1);:
 PRINT "0";: PRINT MID(X$,H3,1);: GOTO 2
080
 2320 IF H3=0 THEN  PRINT MID(X$,H1,1);:
 PRINT MID(X$,H2,1);: PRINT "0";: GOTO 2
080
 2330 PRINT MID(X$,H1,1);: PRINT MID(X$,
H2,1);: PRINT MID(X$,H3,1);: GOTO 2080
```

33 アマチュア無線への応用

I. モールスデコーダ
II. モールスキーイングコレクタ
III. モールスコードランダムジェネレータ

EX-80

清野登代松

I モールス　デコーダ

電鍵で打信したり，受信機で受信したモールスコードを直接文字で表わすことは，マイコン出現のおかげで容易になり，いまでは専用機も市販されるようになりました。

しかし，折角のマイコンキットを活用しない手はないというマニアの方々向けにEX-80を使ったデコードプログラムを開発したのでご紹介します。

EX-80は，インテル型8ビットCPUを使っていますが，このプログラムは，1字分のコードを16ビットで構成しているため$\overline{\mathrm{BT}}$のような特殊文字を除き，すべての文字，数字および記号をハードウエアを変更せずにカバーできます。

1．動作のあらまし

I/Oポートに接続した電鍵のメイクとブレイクの読みとりおよびそれらの時間の長短を識別してコードデータを組立てます。

次にあらかじめストアしたコードテーブルの中からそのデータに対応する文字データをルックアップ（探し出し）し，DMA転送機能を通してTVにディスプレイします。

2．ハードウエア

○EX-80　標準実装0.5KRAM付

○TV受像機

○電　鍵

電鍵は，I/OポートAの任意の1ピンとGNDに直接接続します。

3．プログラムの説明

○メインプログラム

このプログラム専用のフラグ0と1をクリアし，SUB.TVCLRをコールして画面をクリアします。

次にSUB．CHAR1から得た文字データを，SUB．TVDSPを使ってその中で使われるキャラクタ・ジェネレータで文字に変換した上，画面に表示します。

1行10文字が終るとSUB．NLX2をコールして改行，10行の表示が満席になるとSUB，LNSF3，をコールして画面をスクロール（巻き上げ）します。

ここでSUB．TVCLER，同TVDSPおよび同NLX2は，EX-80モニタプログラムROM中のサブルーチンそのもの，SUB．LNSF3は，同じくROM中のSUB．LNSFTを，表示行数を上下に1行ずつ増やすようアレンジしたものです。

《第33－1－1図》メインプログラム

○サブルーチン

＊SUB KEY＊

電鍵がメイクすると，入力データがFFHでなくなるため，インクリメントしても0にならないので，マークであると読みとられ，マークモードのフローが実行されます。

マークの継続時間が基準時間（SUB. DELAYの設定時間×設定繰返し回数）より少なければ，Cレジにはドット符号に対応する01Hを，基準時間以上になれば同じくダッシュ符号に対応する02Hをロードします。

電鍵がブレイクすると，スペースモードにフローが移りCレジ内容をAレジに移してリターンします。

次のコールで，マークモードで述べたと同じ基準時間内に電鍵がメイクすると，その時間はマーク間のセグメントスペースとして識別されますが，これは特にデータとして出力されません。

電鍵のブレイク状態が基準時間以上継続すると，Aレジにキャラクタスペースを表わす00Hをロー

《第33－1－2図》SUB.KEY

ドしてリターンします。

＊SUB. CHAR 1＊

このサブルーチンは，コードデータ組立，ワードスペース送出およびルックアップの3機能からなっています。

コードデータ組立

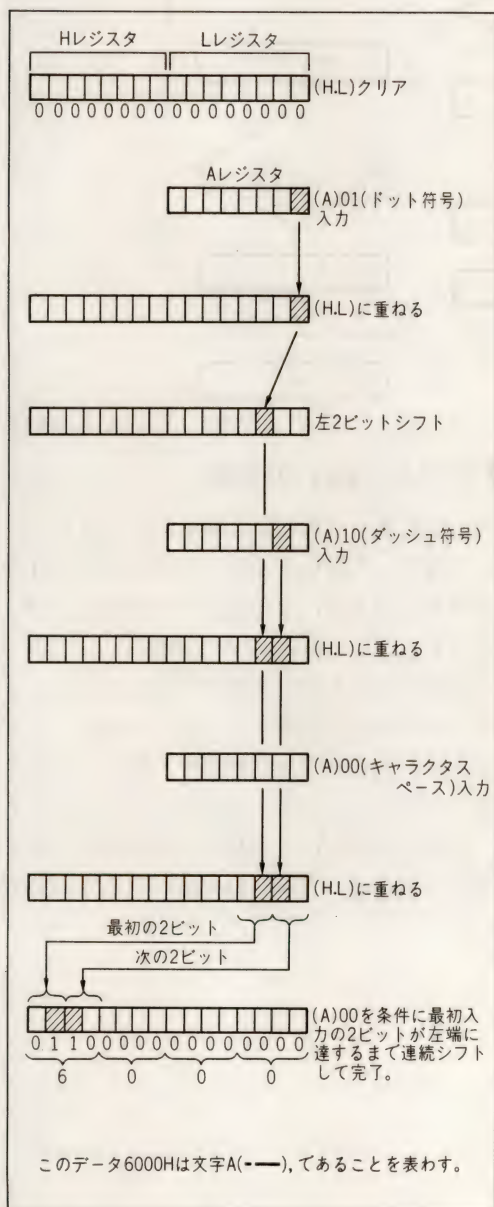
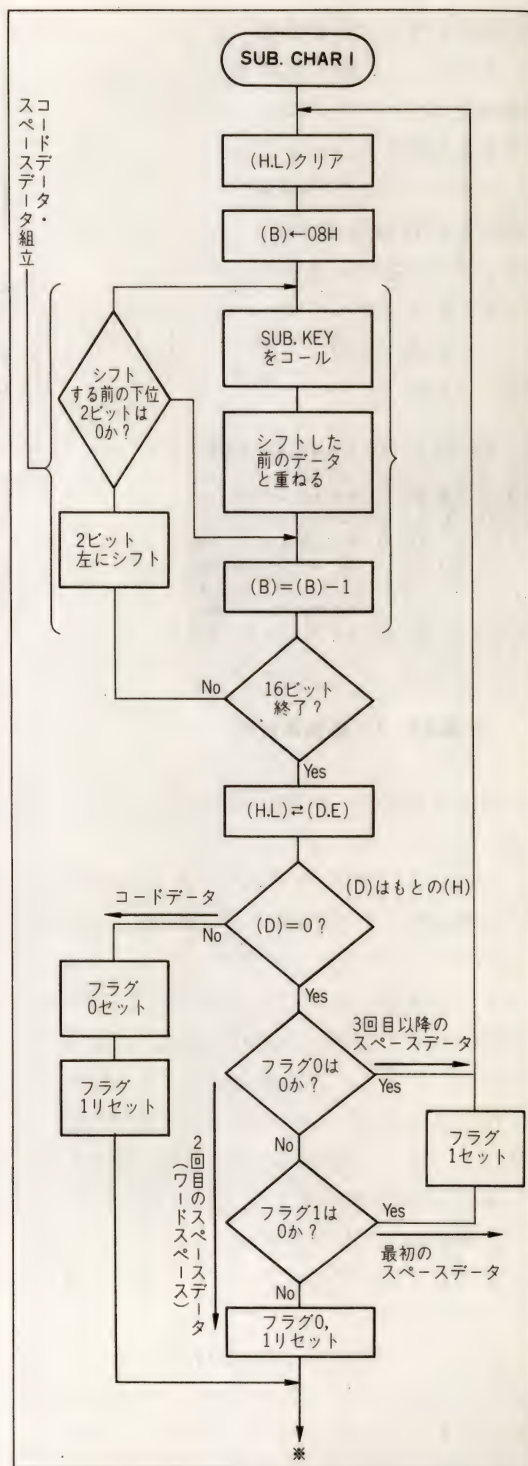
SUB, KEYで得た，ドット，ダッシュの有効ビットは，16ビットのコードデータになるよう最下位ビットから順に2ビットずつ組立てていきます。

キャラクタスペースが入力されると，これで1文

字が終ったことを識別し，有効ビットの最上位ビットが16ビットデータの最上位ビットとなるようシフトして組立てが完了します。

次のコールで再びキャラクタスペースが入力されると16ビットすべてが0からなるスペースデータを作り送出します。

ワードスペース送出

この部分のフローを，コードデータが通過したあと，スペースデータが下がってくると，再びSUB. KEYをコール，ここでまたスペースデータがくると，これを語間すなわちワードスペースとして識別し次のフローに送ります。その後スペースデータが何回入ってこようと，コードデータが入るまでループを保つので，画面はスペース1か所

《第33—1—4図》コードデータ組立て

《第33—1—3図》SUB.CHAR1

をあけたままスタンバイします。

ルックアップ

辞書であるコードテーブルは，82E3H番地から8346H番地まで奇数番地に上位バイト，続く偶数番地に下位バイトのコードをストアしたコードデータテーブルと，8347番地から8378番地まで1バイトずつ文字データをストアした文字データテーブルから成っていて，両データテーブルのコード

《写真33—1》画面実行例

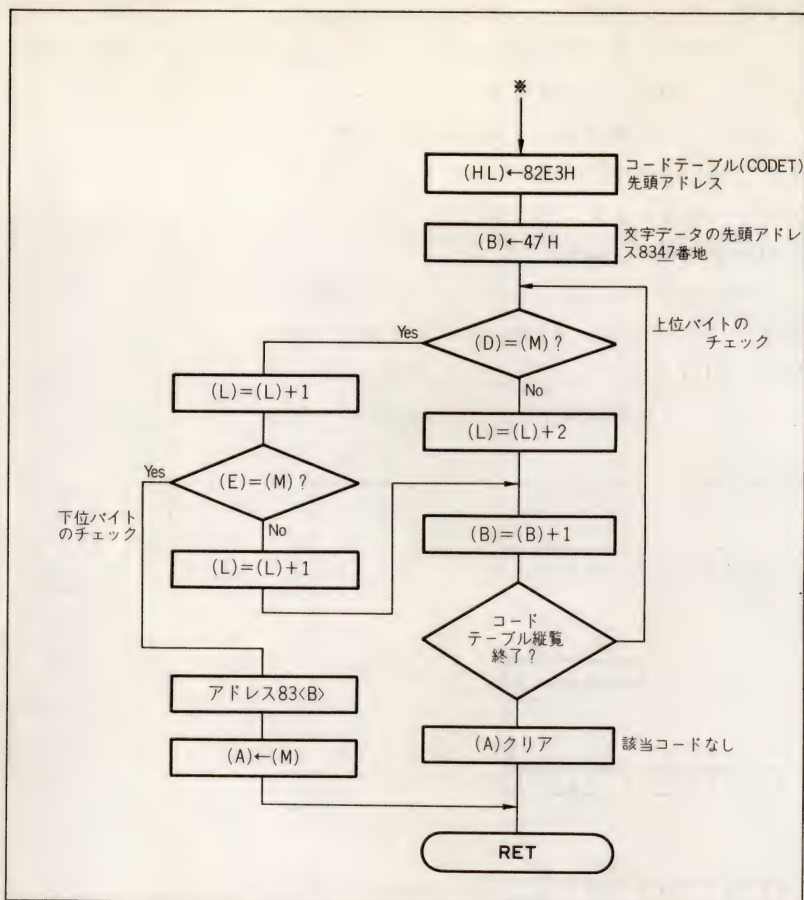
※
(HL)←82E3H　コードテーブル(CODET)先頭アドレス
(B)←47H　文字データの先頭アドレス8347番地
(D)=(M)?　上位バイトのチェック
Yes → (L)=(L)+1
No → (L)=(L)+2
(E)=(M)?　下位バイトのチェック
Yes → アドレス83<B> → (A)←(M) → RET
No → (L)=(L)+1
(B)=(B)+1
コードテーブル縦覧終了?
(A)クリア　該当コードなし
RET

SUB.CHAR1(つづき)《第33—1—5図》

は対応する文字，数字，記号の順序が同一であるよう配列してあります。

ルックアップは，コードデータの上位8ビットと奇数番地のデータを1間とびに照合し，合致したら1番地インクリメントしてコードデータの下位ビットと再び照合，ここでも合致したら1間とびした回数をバイアス値である47に加え，これをLレジにロードします。ここで上位バイトが一致しても下位バイトが違っていれば（たとえば H ，4 ，5 ，訂正）両バイト共一致するまで探します。縦覧が終っても見つからないときは，Aレジに00をロードしてリターンしますが，これをメインプログラムではエラーとして扱い，@で表示します。

ルックアップが終ったら，H，Lペアレジのアドレスデータを手がかりに文字データを拾い出し，これをAレジにロードしてリターンします。

ワードスペースは，厳密には7ドット分ですが，ここではダッシュ単位なので3ダッシュ＝9ドット分になってしまいました。しかし，それほど気にすることはないと思います。また訂正符号の$\overline{HH}$は，標準キャラクタにないのでアステリスク(＊)を当てました。

基準時間すなわち通信速度の自動追従も考えては見たものですが，実際に打つドットやダッシュの長さがムラがあるうちは無意味と思い、固定速度のままとなっています。この速度をかえたいときは、SUB、DELAY中のCVジ内容(82AC番地)がパラメータとなっているので適宜かえてみて下さい。

```
                ;***************************
                ;  MORSE DECODER
                ;***************************
8200                            ORG 8200H
8200    21EA83   BEGIN:  LXI H,FLGRG
8203    3600             MVI M,00H
8205    CD6C03           CALL TVCLR
8208    CD4B82   CHAR0:  CALL CHAR1
820B    CDDE02           CALL TVDSP
820E    3A3E80           LDA DISPX
8211    A7               ANA A
8212    CC6003           CZ NLXZ
8215    CD1B82           CALL LNSF3
8218    C30882           JMP CHAR0
821B    3A3F80   LNSF3:  LDA DISPY
821E    FE14             CPI 14H
8220    C0               RNZ
8221    113080           LXI D,8030H
8224    215080           LXI H,8050H
8227    060B     LNSF1:  MVI B,0BH
8229    7E       LNSF2:  MOV A,M
822A    12               STAX D
822B    23               INX H
822C    13               INX D
822D    05               DCR B
822E    C22982           JNZ LNSF2
8231    011500           LXI B,15H
8234    09               DAD B
8235    EB               XCHG
8236    09               DAD B
8237    EB               XCHG
8238    3E90             MVI A,90H
823A    BD               CMP L
823B    C22782           JNZ LNSF1
823E    3E81             MVI A,81H
8240    BC               CMP H
8241    C22782           JNZ LNSF1
8244    210012           LXI H,1200H
8247    223E80           SHLD DISPX
824A    C9               RET
824B    210000   CHAR1:  LXI H,0000H
824E    0608             MVI B,08H
8250    CDB382   CHAR2:  CALL KEY
8253    B5               ORA L
8254    6F               MOV L,A
8255    05       CHAR3:  DCR B
8256    C28582           JNZ CHAR7
8259    EB               XCHG
825A    21EA83           LXI H,FLGRG
825D    7A               MOV A,D
825E    A7               ANA A

825F    C28F82           JNZ CHAR8
8262    7E               MOV A,M
8263    E601             ANI 01H
8265    CA4B82           JZ CHAR1
8268    7E               MOV A,M
8269    E602             ANI 02H
826B    CA9482           JZ CHAR9
826E    3600             MVI M,00H
8270    21E382   CHAR4:  LXI H,CODET
8273    0647             MVI B,47H
8275    7A       CHAR5:  MOV A,D
8276    BE               CMP M
8277    CA9B82           JZ CHARA
827A    23               INX H
827B    23               INX H
827C    04       CHAR6:  INR B
827D    7D               MOV A,L
827E    FE47             CPI 47H
8280    C27582           JNZ CHAR5
8283    AF               XRA A
8284    C9               RET
8285    29       CHAR7:  DAD H
8286    29               DAD H
8287    E603             ANI 03H
8289    C25082           JNZ CHAR2
828C    C35582           JMP CHAR3
828F    3601     CHAR8:  MVI M,01H
8291    C37082           JMP CHAR4
8294    3E02     CHAR9:  MVI A,02H
8296    B6               ORA M
8297    77               MOV M,A
8298    C34B82           JMP CHAR1
829B    7B       CHARA:  MOV A,E
829C    23               INX H
829D    BE               CMP M
829E    CAA582           JZ CHARB
82A1    23               INX H
82A2    C37C82           JMP CHAR6
82A5    2683     CHARB:  MVI H,83H
82A7    68               MOV L,B
82A8    7E               MOV A,M
82A9    C9               RET
82AA    C5       DELAY:  PUSH B
82AB    0E90             MVI C,90H
82AD    0D       LOOP:   DCR C
82AE    C2AD82           JNZ LOOP
82B1    C1               POP B
82B2    C9               RET
82B3    C5       KEY:    PUSH B
82B4    0600             MVI B,00H
82B6    48               MOV C,B

82B7    DBF8     KEY10:  IN 0F8H
82B9    3C               INR A
82BA    CAD182           JZ KEY30
82BD    CDAA82           CALL DELAY
82C0    04               INR B
82C1    78               MOV A,B
82C2    FE20             CPI 20H
82C4    D2CC82           JNC KEY20
82C7    0E01             MVI C,01H
82C9    C3B782           JMP KEY10
82CC    0E02     KEY20:  MVI C,02H
82CE    C3B782           JMP KEY10
82D1    79       KEY30:  MOV A,C
82D2    A7               ANA A
82D3    C2E182           JNZ KEY40
82D6    CDAA82           CALL DELAY
82D9    04               INR B
82DA    78               MOV A,B
82DB    FE20             CPI 20H
82DD    DAB782           JC KEY10
82E0    AF               XRA A
82E1    C1       KEY40:  POP B
82E2    C9               RET
82E3    6000     CODET:  DW 0060H ;A
82E5    9500             DW 0095H ;B
82E7    9900             DW 0099H ;C
82E9    9400             DW 0094H ;D
82EB    4000             DW 0040H ;E
82ED    5900             DW 0059H ;F
82EF    A400             DW 00A4H ;G
82F1    5500             DW 0055H ;H
82F3    5000             DW 0050H ;I
82F5    6A00             DW 006AH ;J
82F7    9800             DW 0098H ;K
82F9    6500             DW 0065H ;L
82FB    A000             DW 00A0H ;M
82FD    9000             DW 0090H ;N
```

```
82FF    A800        DW 00A8H ;O
8301    6900        DW 0069H ;P
8303    A600        DW 00A6H ;Q
8305    6400        DW 0064H ;R
8307    5400        DW 0054H ;S
8309    8000        DW 0080H ;T
830B    5800        DW 0058H ;U
830D    5600        DW 0056H ;V
830F    6800        DW 0068H ;W
8311    9600        DW 0096H ;X
8313    9A00        DW 009AH ;Y
8315    A500        DW 00A5H ;Z
8317    6A80        DW 806AH ;1
8319    5A80        DW 805AH ;2
831B    5680        DW 8056H ;3
831D    5580        DW 8055H ;4
831F    5540        DW 4055H ;5
8321    9540        DW 4095H ;6
8323    A540        DW 40A5H ;7
8325    A940        DW 40A9H ;8
8327    AA40        DW 40AAH ;9
8329    AA80        DW 80AAH ;0
832B    5555        DW 5555H ;(*)
832D    6660        DW 6066H ;.
832F    A5A0        DW 0A0A5H ;,
8331    A950        DW 50A9H ;:
8333    5A50        DW 505AH ;?
8335    6A90        DW 906AH ;'
8337    9560        DW 6095H ;-
8339    9A40        DW 409AH ;(
833B    9A60        DW 609AH ;)
833D    9640        DW 4096H ;/
833F    9580        DW 8095H ;=
8341    6640        DW 4066H ;+
8343    6590        DW 9065H ;"
8345    0000        DW 0000H;
8347    01          DB 01H ;A
8348    02          DB 02H ;B
8349    03          DB 03H ;C
834A    04          DB 04H ;D
834B    05          DB 05H ;E
834C    06          DB 06H ;F
834D    07          DB 07H ;G
834E    08          DB 08H ;H
834F    09          DB 09H ;I
8350    0A          DB 0AH ;J
8351    0B          DB 0BH ;K
8352    0C          DB 0CH ;L
8353    0D          DB 0DH ;M
8354    0E          DB 0EH ;N
8355    0F          DB 0FH ;O
8356    10          DB 10H ;P
8357    11          DB 11H ;Q
8358    12          DB 12H ;R
8359    13          DB 13H ;S
835A    14          DB 14H ;T
835B    15          DB 15H ;U
835C    16          DB 16H ;V
835D    17          DB 17H ;W
835E    18          DB 18H ;X
835F    19          DB 19H ;Y
8360    1A          DB 1AH ;Z
8361    31          DB 31H ;1
8362    32          DB 32H ;2
8363    33          DB 33H ;3
8364    34          DB 34H ;4
8365    35          DB 35H ;5
8366    36          DB 36H ;6
8367    37          DB 37H ;7
8368    38          DB 38H ;8
8369    39          DB 39H ;9
836A    30          DB 30H ;0
836B    2A          DB 2AH ;(*)
836C    2E          DB 2EH ;.
836D    2C          DB 2CH ;,
836E    3A          DB 3AH ;:
836F    3F          DB 3FH ;?
8370    27          DB 27H ;'
8371    2D          DB 2DH ;-
8372    1B          DB 1BH ;(
8373    1D          DB 1DH ;)
8374    2F          DB 2FH ;/
8375    3D          DB 3DH ;=
8376    2B          DB 2BH ;+
8377    22          DB 22H ;"
8378    20          DB 20H
                ;
0360            NLXZ EQU 0360H
036C            TVCLR EQU 036CH
02DE            TVDSP EQU 02DEH
803E            DISPX EQU 803EH
803F            DISPY EQU 803FH
83EA            FLGRG EQU 83EAH
                      END
NO PROGRAM ERRORS
```

SYMBOL TABLE

```
* 01

A       0007    B       0000    BEGIN   8200 *  C       0001
CHAR0   8208    CHAR1   824B    CHAR2   8250    CHAR3   8255
CHAR4   8270    CHAR5   8275    CHAR6   827C    CHAR7   8285
CHAR8   828F    CHAR9   8294    CHARA   829B    CHARB   82A5
CODET   82E3    D       0002    DELAY   82AA    DISPX   803E
DISPY   803F    E       0003    FLGRG   83EA    H       0004
KEY     82B3    KEY10   82B7    KEY20   82CC    KEY30   82D1
KEY40   82E1    L       0005    LNSF1   8227    LNSF2   8229
LNSF3   821B    LOOP    82AD    M       0006    NLXZ    0360
PSW     0006    SP      0006    TVCLR   036C    TVDSP   02DE
```

II モールス キーイング　コレクタ

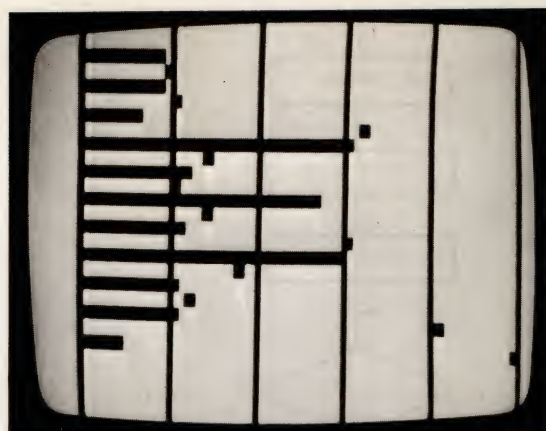
モールスの送信練習をするとき，打ち出したコードが正確かどうかを自分で確められたら……と思うことがありませんか。

アマ国試などでは，コードを紙テープに記録できる試験機を使っていますが，私達がこの機械を練習に利用するわけにはいきません。

そこで，これに似たことをマイコンにやらせようと思い，EX-80向けにプログラムを開発したので紹介します。

1．動作のあらまし

EX-80のビットパターンのDMA転送機能を利用し，I/Oポートに接続した電鍵を操作して，バーと点によるグラフをTVにディスプレイします。

画面にはクリアな状態で時間目盛に用いる6本のタテ線が入っています。この1目盛をモールスコードの1ドット分として利用するとマーク，スペースとも時間長を正確に測ることができ，大変便利です。

《第33—2—1図》TVディスプレイ

《写真33—2》画面表示例

2．ハードウエア

EX-80　標準実装0.5KRAM付

TV受像機

電　鍵

電鍵は，I/OポートAの任意の1ピンとGNDに直接接続します。

3．プログラムの説明

○メインプログラム

まず画面に目盛線を入れ，このプログラムに用いる全フラグをクリアして，KEY入力を待ちます。実際は，電鍵がブレイクの時スペース入力が続々と入ってきているのですが，マーク入力が一度もないときはフラグ2の働きでフローがループになるため，画面が動かないのです。

次に電鍵がメイクすると，入力がFFHでなくなるため，インクリメントした値が0にならないのでフローがマークモードに移ります。

＊マークモード＊

フローがこのモードに移った当初は，フラグ5が0なのでSUB. NEWLNを使って改行，次のSUB. LNSF2をコールして画面が22行埋まったときのスクロールをします。

次にSUB, MARKをコールして，バーの長さを1ビット分伸ばします。

引き続いて各フラグをセット，リセットし，グラフの伸びの速さをきめるSUB. DELAYでウェ

《第33—2—2図》メインプログラム

イトした後，KEYREADに戻ります。このフローは，電鍵がメイクしている間繰り返されるので，バーは繰り返し回数に応じて伸び続けます。

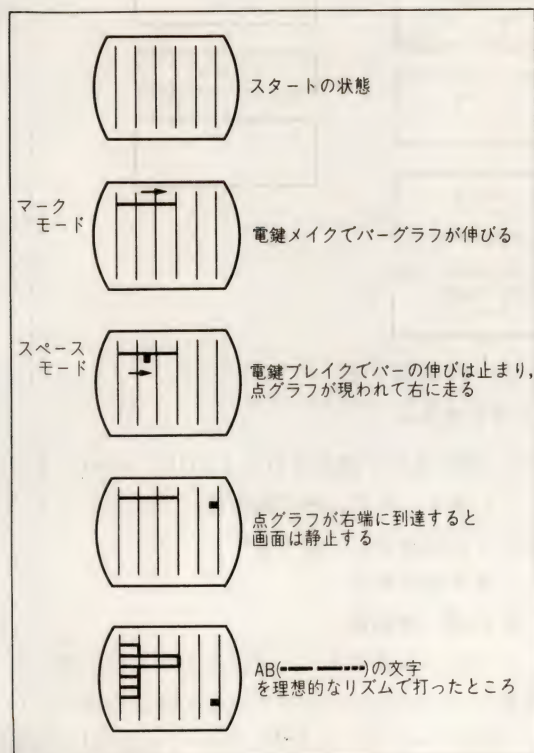
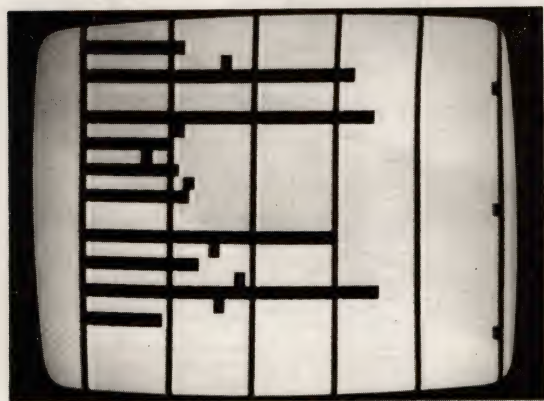
＊スペースモード＊

電鍵がブレイクすると，フローがスペースモードに移ります。

ここで，プログラムRUN後，マークモードの実行を経験したことを表わすフラグ2=1と，点グラフが画面右端にまだ達していないことを示すフラグ0=0の2条件が満足されるとSUB, SPACEをコールします。そのほかの動きは，マークモードの動きとほぼ対応しております。

○サブルーチン

＊SUB．MARK

メインプログラムが，マークモードに切換った当初のコールで，最初の1ビットを画面右端に現わし，次のコールでこれを左に送って右端の1ビットの穴埋めをするというようにしてバーを伸ばしま

《第33－2－3図》フラグのはたらき

《写真33－3》画面表示例

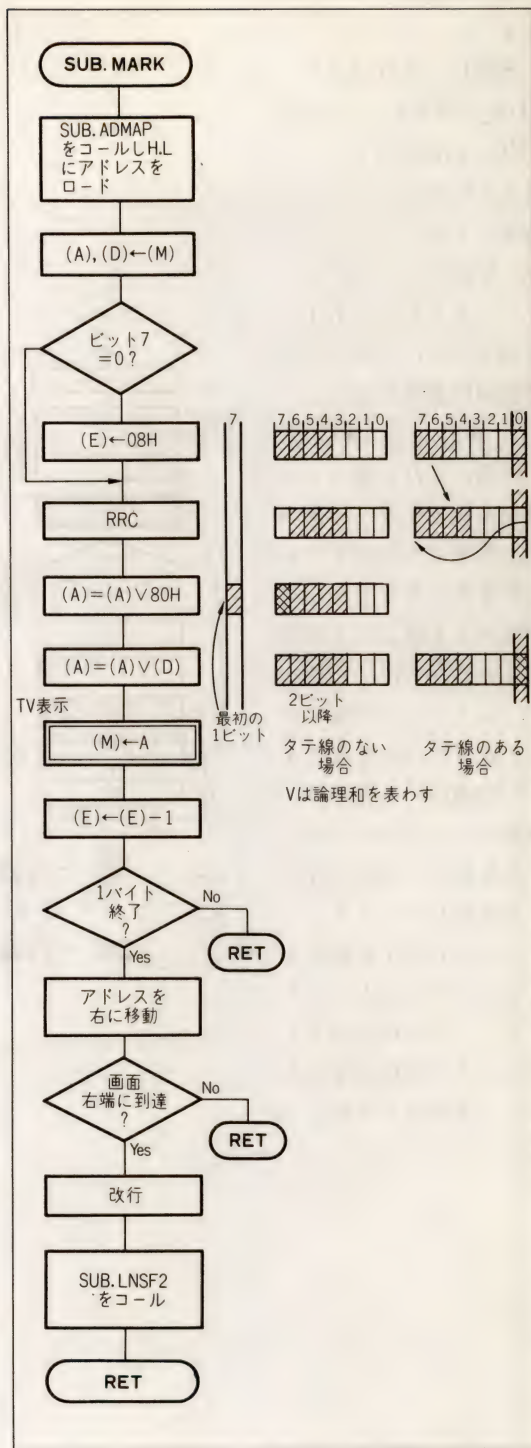
す。1表示アドレスの中にタテ線がある場合は，シフトする前のデータをDレジにセイブし，これをシフト後のデータと重ねてタテ線を残しますが，いずれの場合も同じフローで処理します。

1表示アドレスが満席となったときの右移動，1行全部埋めつくしたときの改行もこのルーチンで行ないます。

＊SUB．SPACE＊

点グラフは，連続した2ビットのパターンが画面を左端から右端に移動するもので，この点と左端のタテ線との距離が，スペースの時間長を表わします。

このルーチンは，バーグラフのときと異なり，2ビット以外の余分のパターンは，タテ線だけを残して消しながら右に移動するのでより複雑なプログラムとなっています。

点が画面右端に達すると，そこで静止し，次のバーグラフとの間にワードスペースを作るためのアドレスだけの改行を行なってリターンします。

《第33－2－4図》SUB.MARK

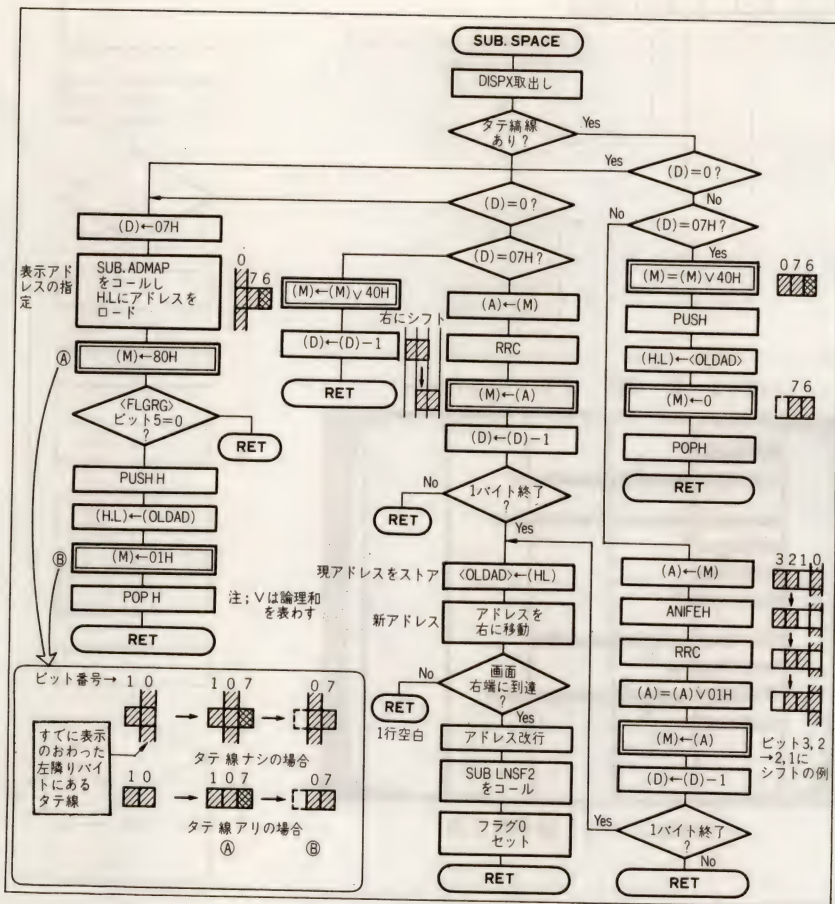
そのほか，各フローの処理と点の移動の様子は，対応させて図示したので，ご理解頂けると思います。

＊その他のサブルーチン＊

SUB. ADMAP, SUB. NEWLN, および SUB. LNSF 2 は, いずれもTVディスプレイに関係のあるルーチンですが, EX-80モニタプログラムにあるものに若干手を加えただけのものなので説明は省略します。

★　　★

実際にこれを使って打鍵してみますと, 自分のクセや手くずれがよくわかります。そして正しいパターンを描こうと練習しているうちに, 自然にモールスコード特有のリズムが身についてきます。さらに面白いことには, 姿勢が正しくないと手くずれを起こしがちになることまでわかります。

これからCWを始めようと思っている方にはもとより, より格調高きオペレータを目指している方にも是非おすすめしたいプログラムです。

《第33—2—5図》SUB.SPACE

```
;*********************************
;*  MORSE  KEYING  CORRECTER  **
;*********************************
8200                    ORG 8200H
8200    211080  BEGIN: LXI H,8010H
8203    060C    CLR10: MVI B,0CH
8205    7D      CLR20: MOV A,L
8206    E601           ANI 01H
8208    C22B82         JNZ CLR40
820B    3601           MVI M,01H
820D    23      CLR30: INX H
820E    05             DCR B
820F    C20582         JNZ CLR20
8212    010400         LXI B,04H
8215    09             DAD B
8216    3EE0           MVI A,0E0H
8218    BD             CMP L
8219    C20382         JNZ CLR10
821C    3E81           MVI A,81H
821E    BC             CMP H
821F    C20382         JNZ CLR10
8222    210000         LXI H,00H
8225    223E80         SHLD DISPX
8228    C33082         JMP RGCLR
822B    3600    CLR40: MVI M,00H
822D    C30D82         JMP CLR30
8230    21A083  RGCLR: LXI H,FLGRG
8233    3600           MVI M,00H
8235    DBF8    INPUT: IN 0F8H
8237    3C             INR A
8238    C26982         JNZ MAIN2
823B    3AA083         LDA FLGRG
823E    4F             MOV C,A
823F    E604           ANI 04H
8241    CA3582         JZ INPUT
8244    79             MOV A,C
8245    E601           ANI 01H
8247    C23582         JNZ INPUT
824A    79             MOV A,C
824B    E608           ANI 08H
824D    C25682         JNZ MAIN1
8250    CD3483         CALL NEWLN
8253    CD4383         CALL LNSF2
8256    CD8D82  MAIN1: CALL SPACE
8259    3AA083         LDA FLGRG
825C    E6DF           ANI 0DFH
825E    F608           ORI 08H
8260    32A083         STA FLGRG
8263    CD8983         CALL DELAY
8266    C33582         JMP INPUT
8269    3AA083  MAIN2: LDA FLGRG

826C    4F             MOV C,A
826D    E620           ANI 20H
826F    C27882         JNZ MAIN3
8272    CD3483         CALL NEWLN
8275    CD4383         CALL LNSF2
8278    CD0883  MAIN3: CALL MARK
827B    3AA083         LDA FLGRG
827E    E6F4           ANI 0F4H
8280    F624           ORI 24H
8282    32A083         STA FLGRG
8285    1600           MVI D,00H
8287    CD8983         CALL DELAY
828A    C33582         JMP INPUT
828D    3A3E80  SPACE: LDA DISPX
8290    E601           ANI 01H
8292    C2E482         JNZ SP40L
8295    7A             MOV A,D
8296    A7             ANA A
8297    CAC782         JZ SP20L
829A    7A             MOV A,D
829B    D607           SUI 07H
829D    CADE82         JZ SP30L
82A0    7E             MOV A,M
82A1    0F             RRC
82A2    77             MOV M,A
82A3    15             DCR D
82A4    C0             RNZ
82A5    22B083  SP10L: SHLD OLDAD
82A8    3A3E80         LDA DISPX
82AB    3C             INR A
82AC    323E80         STA DISPX
82AF    D60A           SUI 0AH
82B1    C0             RNZ
82B2    323E80         STA DISPX
82B5    3A3F80         LDA DISPY
82B8    3C             INR A
82B9    323F80         STA DISPY
82BC    CD4383         CALL LNSF2
82BF    3AA083         LDA FLGRG
82C2    3C             INR A
82C3    32A083         STA FLGRG
82C6    C9             RET
82C7    1607    SP20L: MVI D,07H
82C9    CD7783         CALL ADMAP
82CC    7E             MOV A,M
82CD    F680           ORI 80H
82CF    77             MOV M,A
82D0    3AA083         LDA FLGRG
82D3    E6F0           ANI 0F0H
82D5    C0             RNZ
82D6    E5             PUSH H

82D7    2AB083         LHLD OLDAD
82DA    3601           MVI M,01H
82DC    E1             POP H
82DD    C9             RET
82DE    7E      SP30L: MOV A,M
82DF    F640           ORI 40H
82E1    77             MOV M,A
82E2    15             DCR D
82E3    C9             RET
82E4    7A      SP40L: MOV A,D
82E5    A7             ANA A
82E6    CAC782         JZ SP20L
82E9    7A             MOV A,D
82EA    D607           SUI 07H
82EC    CAFB82         JZ SP50L
82EF    7E             MOV A,M
82F0    E6FE           ANI 0FEH
82F2    0F             RRC
82F3    F601           ORI 01H
82F5    77             MOV M,A
82F6    15             DCR D
82F7    C0             RNZ
82F8    C3A582         JMP SP10L
82FB    7E      SP50L: MOV A,M
82FC    F640           ORI 40H
82FE    77             MOV M,A
82FF    E5             PUSH H
8300    2AB083         LHLD OLDAD
8303    3600           MVI M,00H
8305    E1             POP H
8306    15             DCR D
8307    C9             RET
8308    CD7783  MARK:  CALL ADMAP
830B    7E             MOV A,M
830C    57             MOV D,A
830D    E680           ANI 80H
830F    C21483         JNZ MK10L
8312    1E08           MVI E,08H
8314    7A      MK10L: MOV A,D
8315    0F             RRC
8316    F680           ORI 80H
```

```
8318    B2               ORA D
8319    77               MOV M,A
831A    1D               DCR E
831B    CØ               RNZ
831C    3A3E8Ø           LDA DISPX
831F    3C               INR A
832Ø    323E8Ø           STA DISPX
8323    D6ØA             SUI ØAH
8325    CØ               RNZ
8326    323E8Ø           STA DISPX

8329    3A3F8Ø           LDA DISPY
832C    3C               INR A
832D    323F8Ø           STA DISPY
833Ø    CD4383           CALL LNSF2
8333    C9               RET
8334    3A3E8Ø   NEWLN:  LDA DISPX
8337    AF               XRA A
8338    323E8Ø           STA DISPX
833B    3A3F8Ø           LDA DISPY
833E    3C               INR A
833F    323F8Ø           STA DISPY
8342    C9               RET
8343    D5       LNSF2:  PUSH D
8344    3A3F8Ø           LDA DISPY
8347    FE17             CPI 17H
8349    C27583           JNZ LNSF5
834C    11308Ø           LXI D,8Ø3ØH
834F    21408Ø           LXI H,8Ø4ØH
8352    Ø6ØB     LNSF3:  MVI B,ØBH
8354    7E       LNSF4:  MOV A,M
8355    12               STAX D
8356    23               INX H
8357    13               INX D
8358    Ø5               DCR B
8359    C25483           JNZ LNSF4
835C    Ø1Ø5ØØ           LXI B,Ø5H
835F    Ø9               DAD B
836Ø    EB               XCHG
```

```
8361    Ø9               DAD B
8362    EB               XCHG
8363    3EBØ             MVI A,ØBØH
8365    BD               CMP L
8366    C25283           JNZ LNSF3
8369    3E81             MVI A,81H
836B    BC               CMP H
836C    C25283           JNZ LNSF3
836F    21ØØ16           LXI H,16ØØH
8372    223E8Ø           SHLD DISPX
8375    D1       LNSF5:  POP D
8376    C9               RET
8377    2A3E8Ø   ADMAP:  LHLD DISPX
837A    4D               MOV C,L
837B    6C               MOV L,H
837C    Ø6ØØ             MVI B,ØØH
837E    6Ø               MOV H,B
837F    29               DAD H
838Ø    29               DAD H
8381    29               DAD H
8382    29               DAD H
8383    Ø9               DAD B
8384    Ø1318Ø           LXI B,8Ø31H

8387    Ø9               DAD B
8388    C9               RET
8389    Ø64Ø     DELAY:  MVI B,4ØH
838B    Ø5       LOOP:   DCR B
838C    C28B83           JNZ LOOP
838F    C9               RET
                         ;
                         ;
83AØ             FLGRG EQU 83AØH
83BØ             OLDAD EQU 83BØH
8Ø3E             DISPX EQU 8Ø3EH
8Ø3F             DISPY EQU 8Ø3FH
                         ;
                         END
```

SYMBOL TABLE

```
* Ø1

A       ØØØ7    ADMAP   8377    B       ØØØØ    BEGIN   82ØØ *
C       ØØØ1    CLR1Ø   82Ø3    CLR2Ø   82Ø5    CLR3Ø   82ØD
CLR4Ø   822B    D       ØØØ2    DELAY   8389    DISPX   8Ø3E
DISPY   8Ø3F    E       ØØØ3    FLGRG   83AØ    H       ØØØ4
INPUT   8235    L       ØØØ5    LNSF2   8343    LNSF3   8352
LNSF4   8354    LNSF5   8375    LOOP    838B    M       ØØØ6
MAIN1   8256    MAIN2   8269    MAIN3   8278    MARK    83Ø8
MK1ØL   8314    NEWLN   8334    OLDAD   83BØ    PSW     ØØØ6
RGCLR   823Ø    SP      ØØØ6    SP1ØL   82A5    SP2ØL   82C7
SP3ØL   82DE    SP4ØL   82E4    SP5ØL   82FB    SPACE   828D
```

III モールスコード・ランダム・ジェネレータ

モールス通信の聞き取り練習用に，モールスコードをランダムに発生するプログラムをEX-80用に開発したのでご紹介します。

1．動作のあらまし

乱数を作り，これをアドレスとして，あらかじめメモリにストアしたコードデータをランダムに取り出します。このデータをもとに所定の時間の長さで断続制御された可聴周波のパルスをRF変調し，TVから音声として出力します。

また，実際の通信文を模疑して，5文字を1語にまとめ，語間にワードスペースを入れます。

2．プログラムの説明

○メインプログラム

まず，音程，速度，1語の字数（5）の諸データを，汎用レジスタにロードします。

次に，レジスタRNSD1にストアされている乱数データを取り出し，最下位ビットをマスクして偶数化します。もしこの値が，コードテーブルの最後尾アドレスの上位バイトの値をオーバーしたらSUB, RNGENをコールし，レジスタのデータを入れ替えてやり直します。

乱数がきまるとこれをLレジにロードし，Hレジに83Hをロードしてアドレスとし，コードテーブルから2バイトデータのうちの上位1バイトのコードデータをとり出します。

コードテーブルの格納状況を，数字1の例で示すと次の通りです。

文字	モールスコード	コードテーブル格納状況			
		0110	1010	1000	0000
1	－－－－－－	6	A	8	O
		83B 8 番地にストア		83B 9 番地にストア	

取り出された1バイトデータは，上位から2ビットずつチェックされますが，ここでは右に2回シフトして最下位に移してこれを実行します。

この2ビットが，10_2ならばダッシュ音長のデータ（3）を，01_2ならばドット音長のデータをCレジにロードしてマークルーチンに移ります。マークルーチンの音出しのフローは，音楽演奏プログラムでおなじみのものです。

ただ，ダッシュ，ドットのいずれも音出しのあとに，1ドット長に相当する休みを入れてあるところが音楽とは違います。

この休みが終ると，いま実行した2ビットのデータをクリアします。これは，メインルーチンで実行する2ビットずつのチェックの際，最高4回の2ビットシフトで1バイトのチェックが終了したことを検知させるために行なうものです。

さて，チェックした2ビットが00であると，データをとり出したときのアドレスが偶数であったか，奇数であったかをチェックします。偶数であったら，Lレジをインクリメントし，下位バイトのコードデータを取り出し，上位ビットと同じ処理を行ないます。奇数アドレスであったときは，1字終了ですから，1語中の字数データをデクリメントして1語終了したかどうかをチェックします。

ここで，1語終了してないときは，キャラクタスペース長データをCレジにロード，終了したときは，ワードスペース長データを同じようにロードしてSUB, SPACEをコールし，メインルーチンの始めの部分に戻ります。ここで注意することは，キャラクタ，ワードスペース長のデータは，マークルーチンでドット分のスペースをとってあるの

《第33—3—1図》メインプログラム

で,これを差引いた長さとなることです。

○サブルーチン

＊SUB SPACE＊

このルーチンは,メインプログラムのマークルーチンと音がでないことを除いては同じです。最後にSUB, RNGENをコールしてリターンします。

＊SUB. RNGEN＊

これを乱数と呼ぶにふさわしいかどうか意見の別れるところでしょうが,堅いことは言わないことにします。

このルーチンに用いるレジスタRNSD1, RNSD2にRUNに先立って末尾が0以外の適当な16進数を入れておきます。あとはこの二つの値を足して加数をRNSD1に,和をRNSD2にストアして終ります。これを実際に動かして見ますと,100字以内には同じ繰返しがこないようです。また最初に入れるタネをかえれば,全く別の乱数の組合わせになるので,変化に富んでおり,実用上十分と思います。

3. コードテーブル

ここには,英字,数字,記号およびハム用特殊文字である$\overline{\mathrm{AS}}$などもカバーして収めてあります。

文字Eのような短かいコードは,通信文中で使用頻度が高いと思われるので,多出するよう,いくつものアドレスにスト

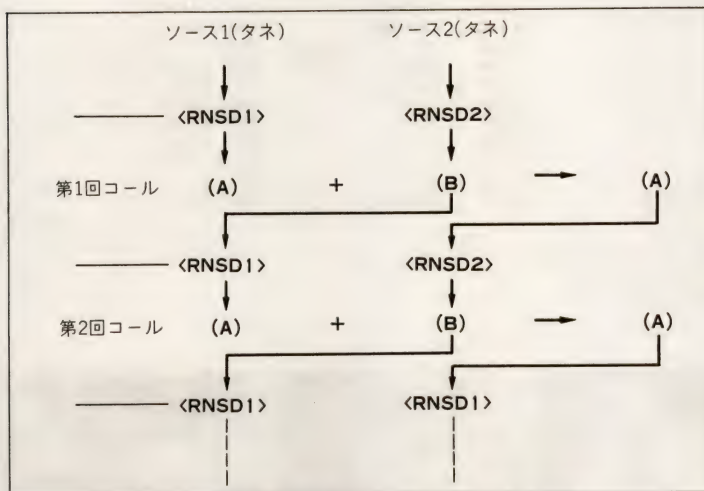
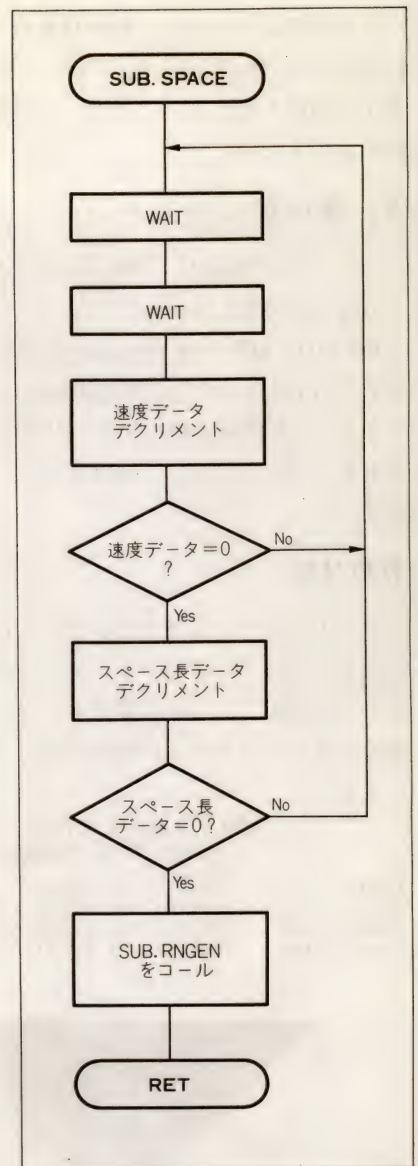
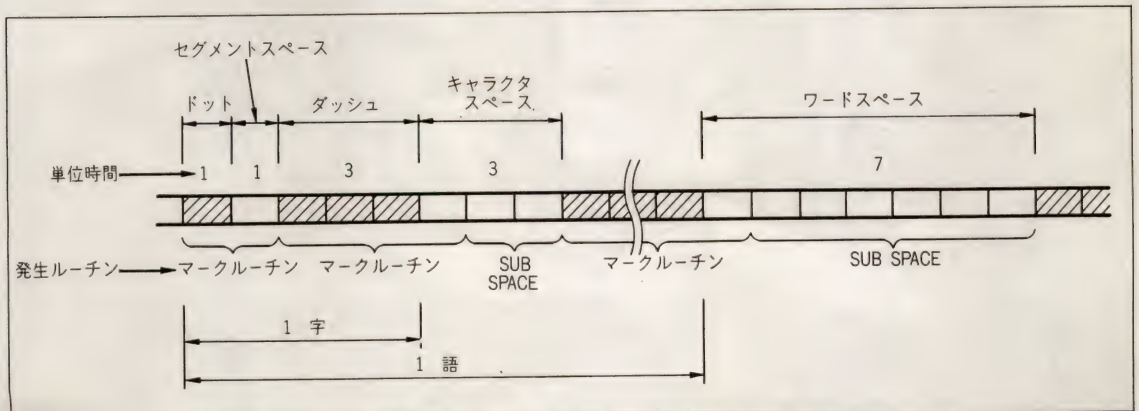
アしました。

なお，この先頭アドレス8300Hは，プログラムの最終82A3Hから跳んでいるので，ロードの際気を付けて下さい。

4．送出速度

速度データは，モールスコードの送出スピードをおおよそきめるもので，これらの関係を下表に示します。

送出速度データ表

速度データ	E6H	80H	60H
送出速度(字/分)	25	45	60
国試レベル	電信アマ	2アマ	1アマ

《第33―3―2図》SUB.RNSDでのデータの動き

《第33―3―4図》SUB.SPACE

《第33―3―3図》モールスコードの発生

プログラムリストは，8203H番地でレジスタDに45字/分に相当するデータ80Hが格納してあります。速度を変えたいときは，この表に従って入れ替えて下さい。

5．使い方

このプログラムは，特別につけ足すハードウエアは何も必要としません。

RNSD1，81F6H番地および2，81F7H番地に任意の下1桁がゼロでない16進数を入れ，ORIGINのアドレス（8200H番地）を頭にRUNすれば，エンドレスにランダムコードがTVスピーカから流れます。

おわりに

コードテーブルにカナコードを入れれば，和文の聞き取り練習に使えます。テープと違って，エンドレスであり，しかも覚えることが不可能な位変数に富んでいるので，実用的なシステムだと思います。

〈プログラム・リスト〉

```
                        ;MORSE CODE RANDOM GENERATOR
                        ;
8200                            ORG 8200H
8200    0625            BEGIN:  MVI B,25H  ;ONTEI & SPEED PARAMETER
8202    1680                    MVI D,80H  ;45CHARACTERS/MIN
8204    1E05            MAIN1:  MVI E,05H  ;5CHARACTERS/WORD
8206    D5              MAIN2:  PUSH D
8207    3AF681                  LDA RNSD1
820A    E6FE            MAIN3:  ANI 0FEH
820C    FEEE                    CPI 0EEH
820E    DA1782                  JC MAIN4
8211    CD8F82                  CALL RNGEN
8214    C30A82                  JMP MAIN3
8217    2683            MAIN4:  MVI H,83H
8219    6F                      MOV L,A
821A    7E              MAIN5:  MOV A,M
821B    07              MAIN6:  RLC
821C    07                      RLC
821D    5F                      MOV E,A
821E    E603                    ANI 03H
8220    CA3182                  JZ MAIN8
8223    3D                      DCR A
8224    CA2C82                  JZ MAIN7
8227    0E03                    MVI C,03H
8229    C34F82                  JMP MARK
822C    0E01            MAIN7:  MVI C,01H
822E    C34F82                  JMP MARK
8231    7D              MAIN8:  MOV A,L
8232    1F                      RAR
8233    DA3A82                  JC MAIN9
8236    2C                      INR L
8237    C31A82                  JMP MAIN5
823A    D1              MAIN9:  POP D
823B    1D                      DCR E
823C    CA4782                  JZ MAINA
823F    0E02                    MVI C,02H
8241    CD7982                  CALL SPACE
8244    C30682                  JMP MAIN2
8247    0E06            MAINA:  MVI C,06H
8249    CD7982                  CALL SPACE
824C    C30482                  JMP MAIN1
824F    D5              MARK:   PUSH D
8250    3E08            MARK1:  MVI A,08H
8252    D3FA                    OUT 0FAH
8254    CD9D82                  CALL WAIT
8257    3E00                    MVI A,00H
8259    D3FA                    OUT 0FAH
825B    CD9D82                  CALL WAIT
825E    15                      DCR D
825F    C25082                  JNZ MARK1
```

```
8262   D1              POP D
8263   D5              PUSH D
8264   0D              DCR C
8265   C25082          JNZ MARK1
8268   CD9D82   MARK2: CALL WAIT
826B   CD9D82          CALL WAIT ;MAKE UP SEGMENT SPACE
826E   15              DCR D
826F   C26882          JNZ MARK2
8272   D1              POP D
8273   7B              MOV A,E
8274   E6FC            ANI 0FCH
8276   C31B82          JMP MAIN6
                ;SUBROUTINE  SPACE
8279   D5       SPACE: PUSH D
827A   CD9D82   SPAC1: CALL WAIT
827D   CD9D82          CALL WAIT ;MAKE UP SEGMENT SPACE
8280   15              DCR D
8281   C27A82          JNZ SPAC1
8284   D1              POP D
8285   D5              PUSH D
8286   0D              DCR C
8287   C27A82          JNZ SPAC1
828A   CD8F82          CALL RNGEN
828D   D1              POP D
828E   C9              RET
                       ;
                       ;
               ;SUBROUTINE RANDOM GEN
828F   C5       RNGEN: PUSH B
8290   21F681          LXI H,RNSD1
8293   7E              MOV A,M
8294   23              INX H
8295   46              MOV B,M
8296   80              ADD B
8297   2B              DCX H
8298   70              MOV M,B
8299   23              INX H
829A   77              MOV M,A
829B   C1              POP B
829C   C9              RET
                       ;
                       ;
                  ;SUBROUTINE  WAIT
                       ;
829D   C5       WAIT:  PUSH B
829E   05       LOOP2: DCR B
829F   C29E82          JNZ LOOP2
82A2   C1              POP B
82A3   C9              RET
                       ;
```

```
                  ;MORSE CODE TABLE

8300                  ORG 8300H
8300   6000   CODET:DW 0060H ;  A
8302   6000         DW 0060H ;  A
8304   6000         DW 0060H ;  A
8306   6000         DW 0060H ;  A
8308   9500         DW 0095H ;  B
830A   9500         DW 0095H ;  B
830C   9500         DW 0095H ;  B
830E   9900         DW 0099H ;  C
8310   9900         DW 0099H ;  C
8312   9400         DW 0094H ;  D
8314   9400         DW 0094H ;  D
8316   9400         DW 0094H ;  D
8318   9400         DW 0094H ;  D
831A   4000         DW 0040H ;  E
831C   4000         DW 0040H ;  E
831E   4000         DW 0040H ;  E
8320   4000         DW 0040H ;  E
8322   4000         DW 0040H ;  E
8324   4000         DW 0040H ;  E
8326   4000         DW 0040H ;  E
8328   4000         DW 0040H ;  E
832A   5900         DW 0059H ;  F
832C   5900         DW 0059H ;  F
832E   5900         DW 0059H ;  F
8330   A400         DW 00A4H ;  G
8332   A400         DW 00A4H ;  G
8334   A400         DW 00A4H ;  G
8336   5500         DW 0055H ;  H
8338   5500         DW 0055H ;  H
833A   5500         DW 0055H ;  H
833C   5500         DW 0055H ;  H
833E   5000         DW 0050H ;  I
8340   5000         DW 0050H ;  I
8342   5000         DW 0050H ;  I
8344   5000         DW 0050H ;  I
8346   5000         DW 0050H ;  I
8348   5000         DW 0050H ;  I
834A   6A00         DW 006AH ;  J
834C   6A00         DW 006AH ;  J
834E   9800         DW 0098H ;  K
8350   9800         DW 0098H ;  K
8352   6500         DW 0065H ;  L
8354   6500         DW 0065H ;  L
8356   6500         DW 0065H ;  L
8358   A000         DW 00A0H ;  M
835A   A000         DW 00A0H ;  M
835C   A000         DW 00A0H ;  M
835E   A000         DW 00A0H ;  M
8360   9000         DW 0090H ;  N
8362   9000         DW 0090H ;  N
```

```
8364   9ØØØ        DW ØØ9ØH ;  N
8366   9ØØØ        DW ØØ9ØH ;  N
8368   9ØØØ        DW ØØ9ØH ;  N
836A   A8ØØ        DW ØØA8H ;  O
836C   A8ØØ        DW ØØA8H ;  O
836E   69ØØ        DW ØØ69H ;  P
8370   69ØØ        DW ØØ69H ;  P
8372   A6ØØ        DW ØØA6H ;  Q
8374   A6ØØ        DW ØØA6H ;  Q
8376   A6ØØ        DW ØØA6H ;  Q
8378   A6ØØ        DW ØØA6H ;  Q
837A   64ØØ        DW ØØ64H ;  R
837C   64ØØ        DW ØØ64H ;  R
837E   64ØØ        DW ØØ64H ;  R
838Ø   64ØØ        DW ØØ64H ;  R
8382   54ØØ        DW ØØ54H ;  S
8384   54ØØ        DW ØØ54H ;  S
8386   54ØØ        DW ØØ54H ;  S
8388   54ØØ        DW ØØ54H ;  S
838A   54ØØ        DW ØØ54H ;  S
838C   8ØØØ        DW ØØ8ØH ;  T
838E   8ØØØ        DW ØØ8ØH ;  T
839Ø   8ØØØ        DW ØØ8ØH ;  T
8392   8ØØØ        DW ØØ8ØH ;  T
8394   8ØØØ        DW ØØ8ØH ;  T
8396   8ØØØ        DW ØØ8ØH ;  T
8398   58ØØ        DW ØØ58H ;  U
839A   58ØØ        DW ØØ58H ;  U
839C   58ØØ        DW ØØ58H ;  U
839E   58ØØ        DW ØØ58H ;  U
83AØ   56ØØ        DW ØØ56H ;  V
83A2   56ØØ        DW ØØ56H ;  V
83A4   56ØØ        DW ØØ56H ;  V
83A6   68ØØ        DW ØØ68H ;  W
83A8   68ØØ        DW ØØ68H ;  W
83AA   68ØØ        DW ØØ68H ;  W
83AC   96ØØ        DW ØØ96H ;  X
83AE   96ØØ        DW ØØ96H ;  X
83BØ   9AØØ        DW ØØ9AH ;  Y
83B2   9AØØ        DW ØØ9AH ;  Y
83B4   A5ØØ        DW ØØA5H ;  Z
83B6   A5ØØ        DW ØØA5H ;  Z
83B8   6A8Ø        DW 8Ø6AH ;  1
83BA   5A8Ø        DW 8Ø5AH ;  2
83BC   568Ø        DW 8Ø56H ;  3
83BE   558Ø        DW 8Ø55H ;  4
83CØ   554Ø        DW 4Ø55H ;  5
83C2   954Ø        DW 4Ø95H ;  6
83C4   A54Ø        DW 4ØA5H ;  7
83C6   A94Ø        DW 4ØA9H ;  8
83C8   AA4Ø        DW 4ØAAH ;  9
```

```
83CA   AA8Ø          DW 8ØAAH ;  Ø
83CC   654Ø          DW 4Ø65H ;  AS
83CE   664Ø          DW 4Ø66H ;  AR
83DØ   958Ø          DW 8Ø95H ;  BT
83D2   9A4Ø          DW 4Ø9AH ;  KN
83D4   566Ø          DW 6Ø56H ;  VA
83D6   666Ø          DW 6Ø66H ;  .
83D8   A5AØ          DW ØAØA5H ;  ,
83DA   A95Ø          DW 5ØA9H ;  :
83DC   5A5Ø          DW 5Ø5AH ;  ?
83DE   6A9Ø          DW 9Ø6AH ;  '
83EØ   956Ø          DW 6Ø95H ;  -
83E2   9A4Ø          DW 4Ø9AH ;  (
83E4   9A6Ø          DW 6Ø9AH ;  )
83E6   964Ø          DW 4Ø96H ;  /
83E8   958Ø          DW 8Ø95H ;  =
83EA   664Ø          DW 4Ø66H ;  +
83EC   659Ø          DW 9Ø65H ;  "
                     ;
81F6          RNSD1 EQU 81F6H ;SOURCE OF RANDOM HEXADECIMAL
81F7          RNSD2 EQU 81F7H ;SOURCE OF RANDOM HEXADECIMAL
                     ;
                     END
NO PROGRAM ERRORS
```

SYMBOL TABLE

```
* Ø1

A       ØØØ7     B       ØØØØ     BEGIN   82ØØ *   C       ØØØ1
CODET   83ØØ *   D       ØØØ2     E       ØØØ3     H       ØØØ4
L       ØØØ5     LOOP2   829E     M       ØØØ6     MAIN1   82Ø4
MAIN2   82Ø6     MAIN3   82ØA     MAIN4   8217     MAIN5   821A
MAIN6   821B     MAIN7   822C     MAIN8   8231     MAIN9   823A
MAINA   8247     MARK    824F     MARK1   825Ø     MARK2   8268
PSW     ØØØ6     RNGEN   828F     RNSD1   81F6     RNSD2   81F7 *
SP      ØØØ6     SPAC1   827A     SPACE   8279     WAIT    829D
```

34

特別効果音回路付き

インベーダ・ゲーム

EX-80BS

市川道教

私はソフト専攻の学生ではありません。マイコンを利用して，測定器等を制御しようと志しています。それが大学生らしくスペ・イン・ブームに巻き込まれ，友人（特に彼女）の要請によって作るはめとなりました。なにぶん不理解のため乱雑なプログラムですが，私のカワユイ，インベーダー君と遊んでやって下さい。

1.プログラムの入力法とメモリーの増設

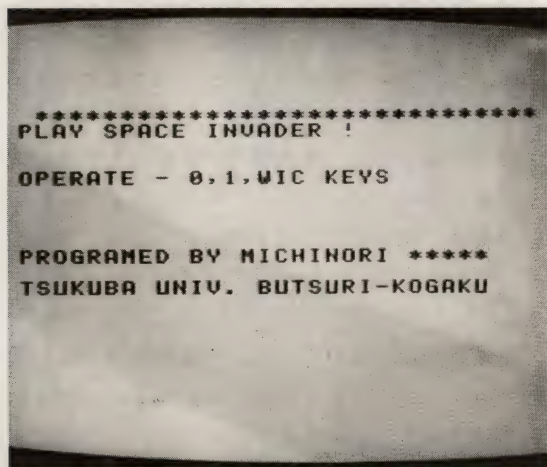
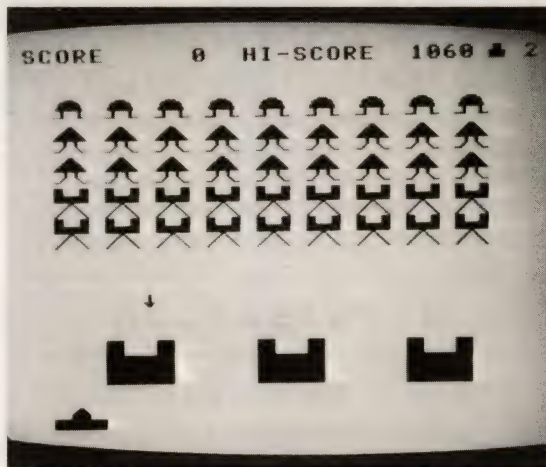
このプログラムを走らせるためには，ＥＸ８０ＢＳにメモリーをＡＢＦＦまで増設する必要があります。その他に一切の改造は不要です。

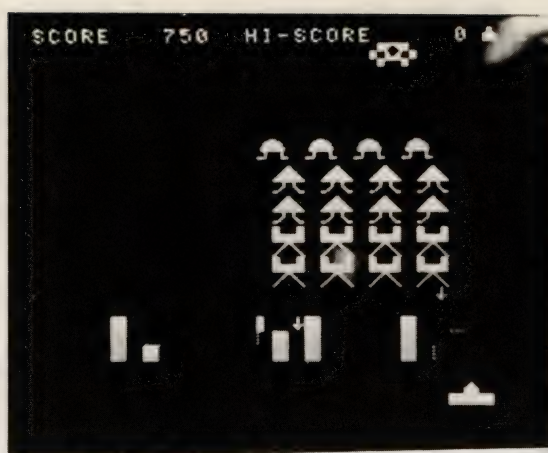
まず，ＢＡＳＩＣをリスト通りに入力して下さい。もっとも２０００番以降のプリント文は省略しても動きます。入力の完了したＢＡＳＩＣプログラムをカセットにストアしておきます。

次にマシン語のダンプ（Ａ０００～Ａ９７０）をそのまま入力して下さい。これも入力が完了したらカセットにストアします。

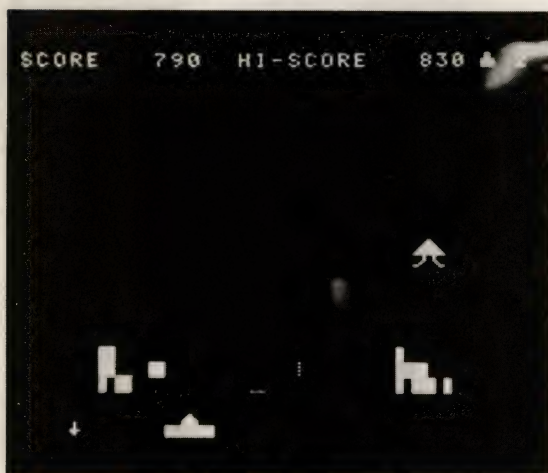
また，秋葉原のマイコンセブンでこのプログラムをデモっていますので，館員の方にお願いしてストアすると楽です。この場合，必ずカセットレコーダも持って行って下さい。

2．RUNのしかたと遊び方

初め，ＢＡＳＩＣのプログラムをＬＯＡＤＨコマンドでカセットからロードします。次にリセット・キーでＥＸ80のモニタにしてから，［ＬＤＡ］

《写真34-1》初期画面

《写真34-2》ゲーム開始

《写真34-3》敵のミサイルで……。

《写真34-4》画面が白黒反転

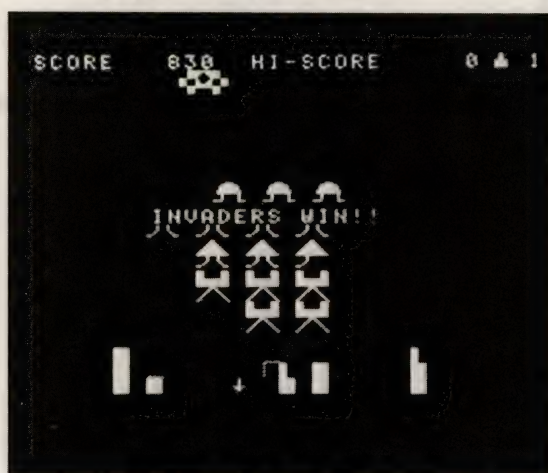
《写真34-5》残るは一ぴき

《写真34-6》インベーダに侵略されて終了

によってマシン語のプログラムをロードします。

カセットの内容がロードされたら，再びBASICモードにします。ここで，RAM・END・ADR，に対してA000と入力して下さい。さあ，あとはRUNと入力すれば，対戦開始！（プログラムの入力に間違いのないときの話ですぞ!!）

コントロールはEX－80の16進キーで行ないます。0で砲台が左，1で右，［WIC］等の青いキーでビームが発生します。たまに，インベーダが半分消えることがありますが，これは，あなたのミスではなく，このプログラムの虫です。でも，なかなかユニークなので直さず放ってあります。

3．プログラムのしくみ

BASICは，画面やレジスタの初期化と得点加算に使い，対戦中の処理のほとんどはマシン語です。

まずは，BASICとマシン語のリンクについて説明します。**第34－1図**を見て下さい。すなわち，マシンからBSに戻るのは，得点された時とインベーダが勝った時の二つの場合です。リプレイは，BS側でインベーダ数が0になった場合に行ないます。Rと言うのは何回目のリプレイかを示し，Rによって最初のインベーダの高さを決めています。

BSのエリアは，まだ相当の余裕がありますから，ここにスター・トレックの小型版でも入れて組み合せてみると言うのも楽しいと思います。参考のため引数は，Sがスコア，Hがハイスコアです。

マシン語は，私自身にとっても難解なぐらい複雑です。そこで，アルゴリズムの思想を参考までに挙げておくことにします。

a）インベーダを動かす

これは，単純でＶＲＡＭ上のあるエリアの内容を左・右に1 byte移すだけです。ただ，その時，ミサイル（Ｄ８）だけは移しません。したがって，さもインベーダだけが動くように見えるのです。下へ動くのも同じです。あとは，この二つの動作を**第34－２図**のようなアルゴリズムでコントロールしています。

b）ビームの発射と命中

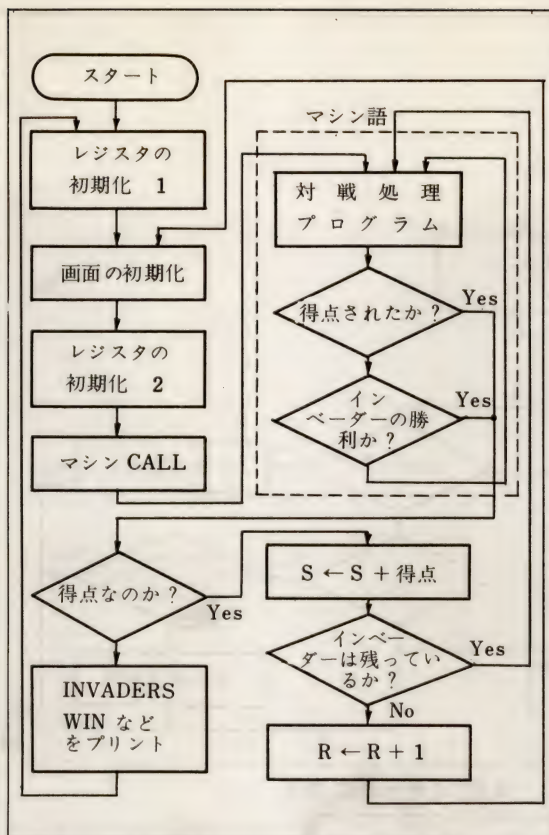
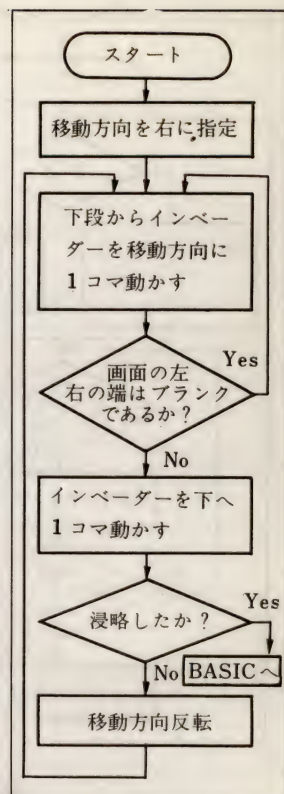
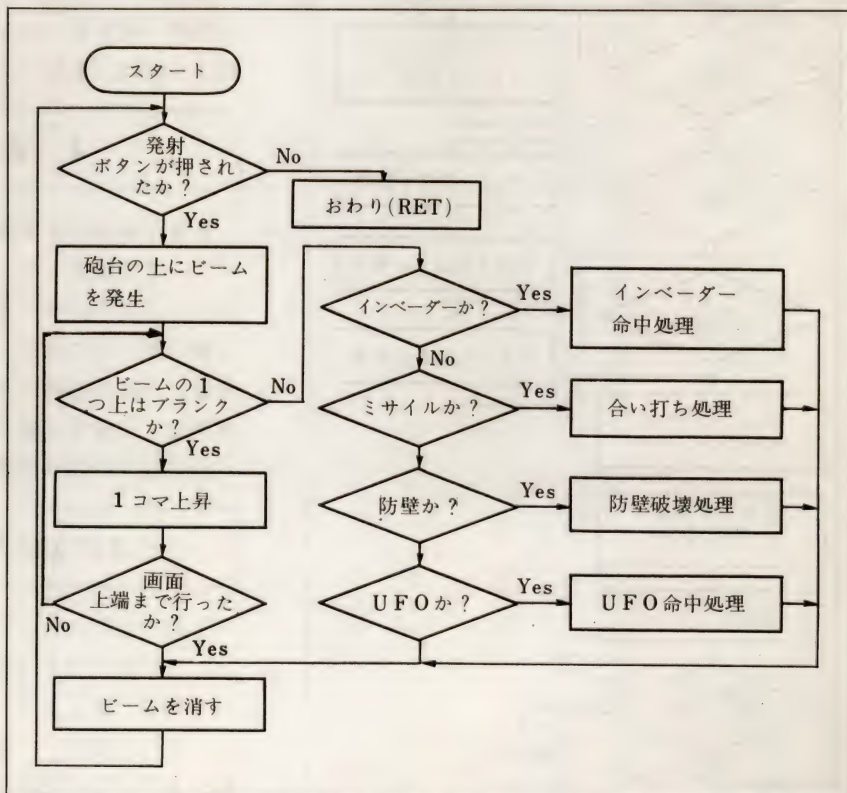
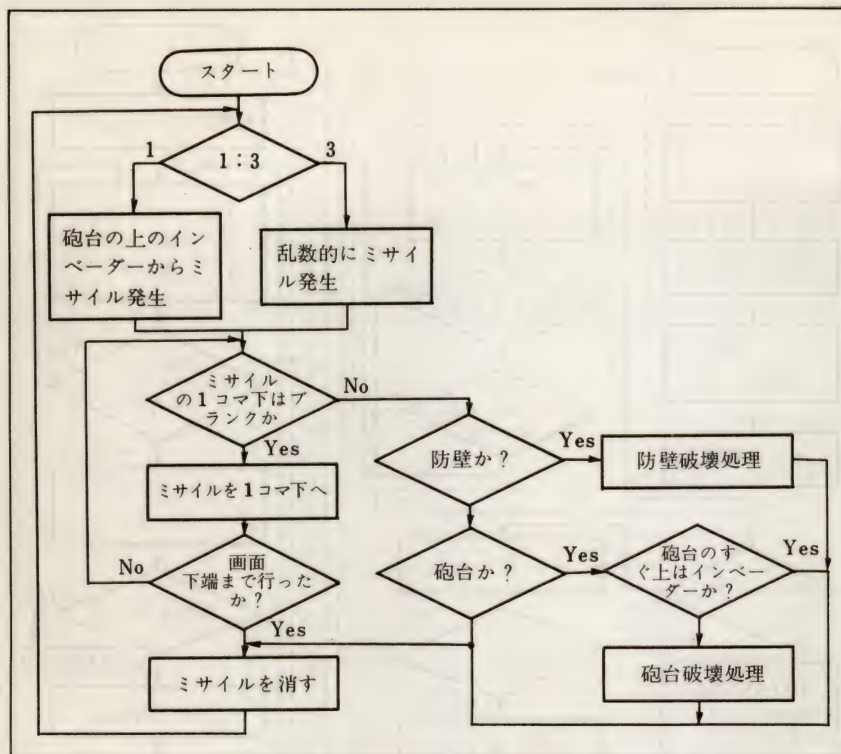
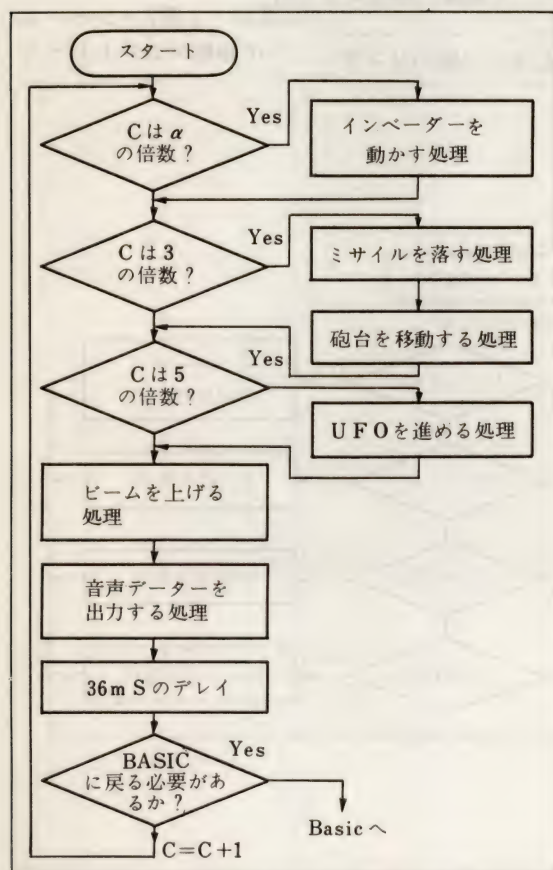
ビームは**第34－３図**のようなアルゴリズムにしたがって行動しています。原理は簡単でビームの１コマ上がブランクならビームを昇上させ，ブランク以外ならば，それに応じた処理をしようと言うだけです（でも，この応じた処理がなかなか大変なのです。）

c）ミサイルの落下と防壁の破壊

ミサイルも原理はビームと全く同様です。**第34－４図**に示されています。ただ，ゲームを楽しくするため三つの工夫をしておきました。一つは，“ナゴヤ打ち”が可能なこと。砲台のすぐ上のインベーダーの放つミサイルは無効です。

二つ目は，防壁で⁝（ＦＢ）はミサイルに負けず，…（ＦＣ）はビームでは壊せません。最後は，合い打ちで，⅗は合い打ち，

《第34－１図》ＢＡＳＩＣとマシン語のリンク

スタート
移動方向を右に指定
下段からインベーダーを移動方向に1コマ動かす
画面の左右の端はブランクであるか？
Yes
No
インベーダーを下へ1コマ動かす
浸略したか？
Yes
No
BASICへ
移動方向反転

《第34－２図》インベーダーの移動のコントロール

《第34－３図》ビームの上昇

《第34－4図》ミサイルの落下

Port B DATA	内　　容
0	インベーダー命中
1	ビーム発射
2	UFO命中
3	UFO進行
4	インベーダー進軍 音程データ
5	インベーダー進軍 音程データ
6	インベーダー進軍
7	あ　き

《第34－1表》音声データ出力

⅕がミサイルの勝ち，⅕がビームの勝ちです。この確率はミサイルの性格で決まり，しつこいミサイルが落ちて来るので注意して下さい。

d）メインプログラム

基本的な思想はタイムシェアリングです。**第34－5図**のようにタイミングに応じて，各種の処理を連続して実行します。タイミングがコールされる処理の速さを決めるのでゆっくりな動きと速い動きが同時に実行できるのです。

《第34－5図》メインプログラムの基本

4．音声発生回路

EX－80の未使用ポートであるPort Bに音声データを出力しています。データはL̇でȮṄになっています。パルスの幅は36m sです。**第34－1表**に割り合てを示しました。**第34－6図**は私の考案した音声回路です。ICを理解なさる方なら，原理はすぐ分ると思います。ハードは苦手な方でも回路通りに作れば動作します。

◁ミスのない入力法とデバッグ▷

1度入力したもののミスを探すのは大変です。ダンプリストの1行分＝16byteを入力する度に戻って確認しましょう。1行の入力と確認が済んだら，リストをペンでチェックして次の行の入力に移ります。この時，画面のアドレスを見てリストと照合するのを忘れないことです。

《第34－6図》音声発生回路

RUNしてみて，ちょっとおかしい時は，ダンプ・リスト横のコメントを頼りにデバッグして下さい（但し，頼りすぎることなかれ！）。

最後に

マイクロコンピュータに接する機会を与えて下さった筑波大，物理工学系増田文男氏及び，本プログラム掲載にご協力下さったマイコンセブン館員の方々に深く感謝します。

プログラム・リスト

```
   1 H=0,R=1;PO.821AH,03H
   2 CALL 0A880H;G.1000
   5 REM SUB.INV INITIAL
   6 G.29
   8 FOR I=1TO2
   9 F.N=1TO9
  10 P."■■ ",  {D0 D1 {
  11 N.N
  12 P." "
  19 F.N=1TO9
  20 P."／＼ ",  {B3 B6 {
  21 N.N
  22 P." "
  25 NEXT I
  26 R.
  29 F.N=1TO9
  30 P."▲ ",  {C2 C3 {
  31 N.N
  32 P." "
  39 F.N=1TO9
  40 P."┘└ ",  {B0 AF {
  41 N.N
  42 P." "
  45 FOR I=1TO 2
  49 F.N=1TO9
  50 P."▲ ",  {B2 B4 {
  51 N.N
  52 P." "
  59 F.N=1TO9
  60 P."ノ＼ ",  {C9 C8 {
  61 N.N
  62 P." "
  65 NEXT I
  70 G.8
 100 CU.X,19;P."■  ■" {80 20 20 80{
 103 CU.X,20;P."■■■■"
 105 CU.X,21;P."■■■■" {80 80 80 80{
 110 R.
 120 X=5;GOS.100
 130 X=14;GOS.100
 140 X=23;GOS.100
 160 R.
 170 P." "
 180 P."  ▲"   {BD {
 190 P." ■■■", {93 80 93 {
 200 R.
 300 PO.8200H,0A0H;PO.8201H,0FCH
 310 PO.8202H,0E0H;PO.8203H,0FCH
 320 PO.8204H,20H;PO.8205H,0FDH
 330 PO.8206H,60H;PO.8207H,0FDH
 340 PO.8208H,0A0H;PO.8209H,0FDH
 350 PO.8213H,02H
 360 PO.8214H,60H;PO.8215H,0FFH
 370 PO.8217H,00H
 380 PO.820AH,09H;PO.820BH,09H
 390 PO.820CH,09H;PO.820DH,09H
 400 PO.820EH,09H
 405 PO.820FH,45H
 410 PO.8219H,00H
 420 PO.8222H,01H
 430 PO.822AH,06H
 440 PO.822BH,15H
 450 PO.8228H,02H
 460 PO.8229H,1AH
 470 PO.8225H,00H
 480 PO.0A65DH,08H
 490 PO.0A69BH,04H
 493 PO.822CH,00H
 495 PO.822DH,0A0H
 496 PO.822EH,0FCH
 510 PO.8232H,40H
 515 PO.8235H,00H
 520 PO.8236H,02H
 530 PO.8237H,02H
 540 PO.809EH,05H
 545 PO.809FH,0A9H
 550 R.
 600 PO.8210H,00H
 605 A=5
 610 PO.8212H,04H
 620 PO.8211H;00H
 630 PO.821CH,03H;PO.821DH,08H
 640 PO.821EH,04H;PO.821FH,01H
 650 CALL 0A630H
 660 IF PE.(8219H)>0 G.1500
 690 IF PE.(8210H) CU.7,10;P."IN
VADERS WIN!!";G.1080
 700 G.650
1000 REM MAIN
1001 R=1
1002 S=0;PO.821AH,03H
1003 Q=0
1004 Z=PE.(821AH)
1005 CL.;P."SCORE",S,"  HI-SCORE
",H," ▲",#2,Z        {DA {
1010 GOS.5
1020 GOS.120
1030 GOS.170
1040 GOS.300
1050 PO.0A004H,82H
1055 PO.0A034H,82H
1060 PO.0A003H,00H
1065 PO.0A033H,00H
1066 FOR J=1TOR
1067 CALL 0A05EH;CALL 0A05EH
1068 N.J
1070 G.600
1080 F.I=1TO5000;N.I
1090 IF S>H H=S
1200 G.1990
1500 S=S+PE.(8219H)*10
1510 IF S>1500 G.1560
1520 CU.0,0;P."SCORE",S
1530 PO.8219H,00H
1540 CALL 0A41DH
1545 IF PE.(820FH)=0 R=R+1;G.100
4
1547 G.650
1560 IF Q=1 G.1520
1570 Q=PE.(821AH)
1580 Q=Q+1
1590 PO.821AH,Q
1600 CU.30,0
1610 P.#1,Q;Q=1;G.1520
1990 CL.;CU.1,4
2000 F.I=1TO30;P."*",;F.N=1TO10;
N.N;N.I;P.
2005 P."PLAY SPACE INVADER !"
2010 F.I=1TO200;N.I
2020 P.;P.;P."OPERATE - 0,1,WIC
KEYS"
2025 F.I=1TO200;N.I
2026 P.;P.
2027 P.;P.;P."PROGRAMED BY MICHI
NORI *****"
2028 P.;P."TSUKUBA UNIV. BUTSURI
-KOGAKU"
2030 F.I=1TO5000
2035 N.I
2040 PO.809EH,10H
2060 CALL 3F70H
2070 PO.809EH,05H
2100 G.1000
OK
>_
```

マシン語　プログラム―ダンプ・リスト

1〈A000～A055〉インベーダーを左右に動かす。

2〈A05E～A102〉インベーダーを下に動かす。

3〈A105～A172〉インベーダーを動かすアルゴリズム。

4〈A180～A1CC〉砲台を動かす。

5〈A1D0～A1E9〉ビームの発射

6〈A1EA～A228〉ビームの上昇

7〈A22C～A234〉HL←HL－20H

8〈A236～A24F〉6の修正プログラム

9〈A260～A299〉インベーダ　命中の際の破壊処理，×にする。

10〈A2A0～A379〉ビームが何かに当った時の処理・分岐

11〈A380～A3B7〉10点インベーダ命中

12〈A3B8～A3BC〉HL←HL＋20H

13〈A3C0～A3F7〉20点インベーダ命中

14〈A3F8～A409〉30点インベーダ命中

15〈A410～A45D〉×を消す，インベーダ総数を計算する。

16〈A45E～A479〉インベーダの動く順を判断

17〈A480～A4C6〉砲台の上からミサイルを発射させる。

Adr. → DATA

```
#PRB 0000 A000 A980
#A000 06 00 2A 00 82 7C C6 00 C8 23 7E 4F D6 D8 C2 14
#A010 A0 0E 20 00 CD FC A8 2B 71 23 78 D6 3E C2 22 A0
#A020 36 20 04 78 D6 3F FA 09 A0 C9 FF 7F 00 00 01 00
#A030 06 00 2A 00 82 7C C6 00 C8 11 3F 00 19 2B 7E 4F
#A040 D6 D8 C2 48 A0 0E 20 00 CD FC A8 23 71 2B 04 78
#A050 D6 40 FA 3D A0 C9 C9 20 20 20 EF FE 40 00 3E 00
#A060 2A 00 82 44 4D 00 21 09 82 56 82 C2 73 A0 2B 2B
#A070 C3 69 A0 2B 5E EB 11 3F 00 19 00 00 00 56 7A D6
#A080 D8 CA A0 A0 7A D6 8C CA A0 A0 7D C6 20 6F 7C CE
#A090 00 67 00 00 72 7D D6 20 6F 7C DE 00 67 00 00 00
#A0A0 7C 90 C2 A8 A0 7D 91 CA AE A0 2B C3 7D A0 06 00
#A0B0 2A 00 82 7E D6 8C CA BB A0 36 20 04 23 78 D6 20
#A0C0 C2 B3 A0 00 01 20 00 16 00 7A 32 D1 A0 32 DB A0
#A0D0 2A 08 82 3E 00 84 CA DD A0 09 22 08 82 14 14 7A
#A0E0 D6 0A C2 C9 A0 00 21 09 82 3E 00 86 C2 F4 A0 2B
#A0F0 2B C3 E9 A0 7E D6 FF C0 2B 7E D6 40 C0 3E 01 32
#A100 10 82 C9 20 20 3A 12 82 47 3D FC 5E A4 00 00 32
#A110 12 82 78 80 32 03 A0 32 33 A0 CD AA A8 C6 00 C2
#A120 28 A1 CD 30 A0 C3 2B A1 CD 00 A0 3A 22 82 A7 C0
#A130 CD 75 A4 3A 11 82 A7 CA 43 A1 21 A0 FC 11 40 FF
#A140 C3 49 A1 21 3F FC 11 5F FF 01 20 00 7C 92 C2 54
#A150 A1 7D 93 C8 7E 91 CA 6F A1 3A 11 82 C6 00 C2 66
#A160 A1 3E 01 C3 68 A1 3E 00 32 11 82 CD 5E A0 C9 09
#A170 C3 4C A1 20 20 20 40 10 3F FF 20 20 20 20 20 20
#A180 3E 65 CD 59 3F 47 C6 00 C8 3D C2 92 A1 00 CD A0
#A190 A1 C9 78 D6 02 C0 00 00 CD B8 A1 C9 20 20 20 20
#A1A0 3A 13 82 47 D6 04 F8 05 78 32 13 82 3E 14 32 03
#A1B0 A0 CD 00 A0 C9 20 00 23 3A 13 82 47 D6 1C F0 04
#A1C0 78 32 13 82 3E 14 32 33 A0 CD 30 A0 C9 20 20 20
#A1D0 3A 17 82 A7 C0 3E 35 CD 36 A2 A7 C8 CD 89 A8 C6
#A1E0 40 6F 26 FF 22 16 82 36 8C C9 3A 17 82 A7 C8 00
#A1F0 2A 16 82 7E D6 8C CA 02 A2 23 7E D6 8C CA 02 A2
#A200 2B 2B 36 20 2A 16 82 CD 2C A2 00 7E D6 20 C2 A0
#A210 A2 7C D6 FC C2 23 A2 7D D6 20 FA 23 A2 3E 00 32
#A220 17 82 C9 36 8C 22 16 82 C9 C3 1D A2 7D D6 20 6F
#A230 7C DE 00 67 C9 FA CD 59 3F A7 C2 44 A2 3C 32 18
#A240 82 3D C9 00 3A 18 82 A7 C8 3E 00 32 18 82 3C C9
#A250 AF DF 0F FF 14 20 8C 10 FF EF 6B EF 00 20 E4 90
#A260 36 FE 2B 36 FD CD 2C A2 36 FE 23 36 FD C9 36 FD
#A270 2B 36 FE 01 20 00 09 36 FD 23 36 FE C9 36 FE 23
#A280 36 FD 01 20 00 09 36 FE 2B 36 FD C9 36 FD 23 36
#A290 FE CD 2C A2 36 FD 2B 36 FE C9 20 DD 00 00 04 20
#A2A0 7E D6 B6 C2 AC A2 CD 60 A2 C3 80 A3 7E D6 D1 C2
#A2B0 B8 A2 CD 6E A2 C3 80 A3 7E D6 D0 C2 C4 A2 CD 7D
#A2C0 A2 C3 80 A3 7E D6 B3 C2 D0 A2 CD 8C A2 C3 80 A3
#A2D0 7E D6 C8 C2 DC A2 CD 60 A2 C3 C0 A3 7E D6 B4 C2
#A2E0 E8 A2 CD 6E A2 C3 C0 A3 7E D6 B2 C2 F4 A2 CD 7D
#A2F0 A2 C3 C0 A3 7E D6 C9 C2 00 A3 CD 8C A2 C3 C0 A3
#A300 7E D6 AF C2 0C A3 CD 60 A2 C3 F8 A3 7E D6 C3 C2
#A310 18 A3 CD 6E A2 C3 F8 A3 7E D6 C2 C2 24 A3 CD 7D
#A320 A2 C3 F8 A3 7E D6 B0 C2 30 A3 CD 8C A2 C3 F8 A3
#A330 7E D6 D8 C2 42 A3 3A 1D 82 3D CA 1D A2 CD B8 A3
#A340 36 B9 7E D6 AB C2 4B A3 C3 D0 A7 7E D6 96 CA 48
#A350 A3 7E D6 AC CA 48 A3 7E D6 80 C2 62 A3 36 82 C3
#A360 1D A2 7E D6 82 C2 6E A3 00 36 FB C3 1D A2 7E D6
#A370 FB C2 79 A3 36 20 C3 1D A2 C3 1D A2 11 20 00 C9
#A380 EB 2A 08 82 EB 7D A7 CA A5 A3 7C 92 FA A5 A3 7D
#A390 93 FA A5 A3 3A 0E 82 3D 32 0E 82 A7 C2 B3 A3 32
#A3A0 09 82 C3 B3 A3 3A 0D 82 3D 32 0D 82 A7 C2 B3 A3
#A3B0 32 07 82 3E 01 C3 01 A4 11 20 00 19 C9 40 40 00
#A3C0 EB 2A 04 82 EB 7D A7 CA E5 A3 7C 92 FA E5 A3 7D
#A3D0 93 FA E5 A3 3A 0C 82 3D 32 0C 82 A7 C2 F3 A3 32
#A3E0 05 82 C3 F3 A3 3A 0B 82 3D 32 0B 82 A7 C2 F3 A3
#A3F0 32 03 82 3E 02 C3 01 A4 3A 0A 82 3D 32 0A 82 3E
 ]
 [
#A400 03 32 19 82 22 20 82 C3 1D A2 F7 FF 0A 0D 20 20
#A410 7E D6 FE CA 1A A4 7E D6 FD C0 36 20 C9 2A 20 82
#A420 CD 10 A4 23 CD 10 A4 2B 2B CD 10 A4 11 20 00 19
#A430 CD 10 A4 23 CD 10 A4 23 CD 10 A4 CD 2C A2 CD 2C
#A440 A2 CD 10 A4 2B CD 10 A4 2B CD 10 A4 06 04 0E 00
#A450 21 0A 82 7E 81 4F 23 05 C2 53 A4 23 71 C9 3E 00
#A460 32 22 82 0E 04 21 09 82 7E A7 C2 73 A4 2B 2B 0D
#A470 C3 68 A4 79 C9 3E 01 32 22 82 C9 3A 31 82 47 00
#A480 3A 25 82 A7 C0 0E 03 3A 13 82 C6 60 6F 06 0E 26
#A490 FF CD 2C A2 CD 88 A5 A7 CA BD A4 05 C2 91 A4 3A
#A4A0 13 82 C6 3E 6F 0D 00 79 D6 02 CA 8D A4 2C 79 D6
#A4B0 01 CA 8D A4 2C 2C 2C 79 A7 CA 8D A4 C9 11 20 00
```

- 18《A4C8～A4E6》 ランダムにミサイル発射
- 19《A4F0～A584》 ミサイルの落下と防壁の破壊処理
- 20《A588～A59F》 インベーダのチェック
- 21《A5A9～A5E6》 砲台の破壊
- 22《A5E8～A60B》 砲台の発生
- 23《A630～A6B0》 マシン　メイン!!
- 24《A6C0～A6DD》 ミサイル用ランダム数の発生
- 25《A6E0～A70F》 インベーダーの動く速さの決定
- 26《A715～A7A7》 UFOの発生と移動
- 27《A7AA～A7CF》 UFOの消去
- 28《A7D0～A840》 UFOに命中　10でコールされる。
- 29《A850～A875》 音声データーをポートへ出力，得点時の音声データ
- 30《A880～A888》 I／O　ポート　コマンド　出力
- 31《A889～A8FB》 音声データ　出力
- 32《A905～A918》 音符データ
- 33《A91B～A96B》 ミサイル発生　2

```
#A4C0 19 7E D6 20 C0 C3 09 A5 3A 25 82 A7 C0 0E 04 11
#A4D0 28 82 1A 00 00 C6 60 6F 26 FF 06 0F CD 2C A2 CD
#A4E0 88 A5 A7 CA BD A4 05 C2 DC A4 13 0D C2 D2 A4 C9
#A4F0 0A 6F 03 0A C6 00 C8 67 7E D6 D8 CA 0F A5 7E D6
#A500 20 CA 0F A5 C3 64 A5 00 00 22 24 82 36 D8 C9 36
#A510 20 11 20 00 19 7C D6 FF C2 25 A5 7D C6 20 D2 25
#A520 A5 AF 02 C9 00 7E D6 20 C2 34 A5 36 D8 7C 02 0B
#A530 7D 02 C9 00 7E D6 B9 C2 3E A5 02 36 20 C9 7E D6
#A540 8C C2 46 A5 02 C9 7E D6 80 C2 50 A5 02 36 82 C9
#A550 7E D6 82 C2 5A A5 02 36 FC C9 7E D6 FC C2 64 A5
#A560 02 36 20 C9 7E D6 BD C2 6F A5 02 CD A9 A5 C9 7E
#A570 D6 93 CA 6A A5 AF 02 C9 01 24 82 CD F0 A4 01 26
#A580 82 CD F0 A4 C9 C9 00 00 7E D6 B3 C8 7E D6 B6 C8
#A590 7E D6 C9 C8 7E D6 C8 C8 7E D6 B0 C8 7E D6 AF C9
#A5A0 7E D6 C8 C8 7E D6 C9 C9 00 00 00 C3 52 A9 C6 60
#A5B0 5F 16 FF 6B 62 CD 6E A2 21 0A 00 00 CD 70 3F 00
#A5C0 6B 62 CD 7D A2 21 0A 00 D5 CD D4 02 D1 1D 13 00
#A5D0 00 6B 62 CD 20 A4 3A 1A 82 3D 32 1A 82 A7 C2 E7
#A5E0 A5 3E 01 32 10 82 C9 3E BD 32 63 FF 21 82 FF 36
#A5F0 93 23 36 80 23 36 93 3E 03 32 13 82 21 9E FC 3A
#A600 1A 82 C6 30 77 21 90 00 CD D4 02 C9 00 00 00 00
#A610 36 20 C3 15 A5 11 00 04 F7 F3 FB 75 0F 02 04 08
#A620 F8 F7 F8 FF 09 0D 1D 0B F3 F8 FD FB 0C 0B 0C 0B
#A630 F5 C5 D5 E5 2A 1C 82 EB 2A 1E 82 7A A7 C2 50 A6
#A640 D5 E5 CD 80 A1 CD 78 A5 CD C0 A6 00 E1 D1 16 03
#A650 7B A7 C2 5F A6 D5 E5 CD 05 A1 E1 D1 1E 04 00 7C
#A660 A7 C2 6E A6 D5 E5 CD 1B A9 E1 D1 26 04 00 7D A7
#A670 C2 7E A6 D5 E5 CD C8 A4 E1 D1 2E 05 00 00 7D 00
#A680 D6 02 C2 8E A6 D5 E5 00 00 CD 15 A7 E1 D1 15 1D
#A690 25 2D D5 E5 CD E0 A6 CD D0 A1 21 04 00 CD 50 A8
#A6A0 E1 D1 3A 19 82 00 A7 C2 B1 A6 3A 10 82 A7 CA 3B
#A6B0 A6 22 1E 82 00 EB 22 1C 82 E1 D1 C1 F1 C9 20 20
#A6C0 21 28 82 3A 13 82 86 83 4F 16 04 79 81 4F D6 1D
#A6D0 5F C6 E2 DA CE A6 1C 73 23 15 C2 CB A6 C9 20 20
#A6E0 CD EA A1 21 5D A6 3A 0F 82 47 D6 20 F0 36 07 78
#A6F0 D6 10 F0 36 04 78 D6 04 F0 36 02 05 C0 00 36 01
#A700 21 9B A6 36 03 3A 0A 82 A7 C0 36 02 CD 05 A1 C9
#A710 F1 7E FA F6 0B CD AA A7 3A 0F 82 D6 08 FA 60 A7
#A720 3A 32 82 A7 C2 60 A7 3E 60 32 32 82 3A 11 82 32
#A730 2F 82 A7 CA 3C A7 32 2C 82 C3 41 A7 3E 1C 32 2C
#A740 82 C6 A0 6F 26 FC 36 A3 23 36 CB 23 36 A2 11 20
#A750 00 19 36 AC 2B 36 96 2B 36 AB 00 00 00 00 00 00
#A760 3D 32 32 82 3A 2C 82 A7 C8 21 2D 82 22 03 A0 22
#A770 33 A0 CD 95 A8 A7 C2 88 A7 CD 00 A0 3A 2C 82 3D
#A780 32 2C 82 3D CA 95 A7 C9 CD 30 A0 3A 2C 82 3C 32
#A790 2C 82 D6 1C C0 3A 2C 82 32 34 82 AF 32 2C 82 3E
#A7A0 40 32 32 82 CD B3 A7 C9 FA FA 3A 33 82 3D 32 33
#A7B0 82 A7 C0 3A 34 82 C6 A0 6F 26 FC 06 20 70 23 70
#A7C0 23 70 11 20 00 19 70 2B 70 2B 70 AF 32 33 82 C9
#A7D0 CD A1 A8 26 FC 3A 2C 82 32 34 82 C6 C0 6F AF 32
#A7E0 2C 82 3E 04 32 33 82 3A 1F 82 3D C2 00 A8 3E 05
#A7F0 32 19 82 36 20 23 36 35 23 36 30 C3 35 A8 00 00

[
#A800 3D C2 14 A8 3E 0A 32 19 82 36 31 23 36 30 23 36
#A810 30 C3 35 A8 3D C2 28 A8 3E 0F 32 19 82 36 31 23
#A820 36 35 23 36 30 C3 35 A8 3E 1E 32 19 82 36 33 23
#A830 36 30 23 36 30 CD 2C A2 36 0F 2B 36 06 2B 36 15
#A840 C3 1D A2 FD 11 C0 34 52 7F 7B 37 FE 30 A2 F0 80
#A850 3A 19 82 A7 CA 5F A8 3A 35 82 C6 01 32 35 82 3E
#A860 90 00 00 3A 35 82 EE FF D3 F9 AF 32 35 82 CD D4
#A870 02 3E FF D3 F9 C9 1F F2 00 42 0E 85 70 F3 7B ED
#A880 3E 90 D3 FB 3E FF D3 F9 C9 3A 35 82 C6 02 32 35
#A890 82 3A 13 82 C9 3A 35 82 C6 08 32 35 82 3A 2F 82
#A8A0 C9 3A 35 82 C6 04 32 35 82 C9 C5 D5 3A 36 82 3D
#A8B0 CA B9 A8 32 36 82 C3 F6 A8 3E 04 32 36 82 3A 35
#A8C0 82 47 3A 37 82 57 A7 C2 D1 A8 78 C6 70 4F C3 ED
#A8D0 A8 3D C2 DC A8 78 C6 60 4F C3 ED A8 3D C2 E7 A8
#A8E0 78 C6 50 4F C3 ED A8 78 C6 40 4F 16 FF 14 7A 32
#A8F0 37 82 79 32 35 82 D1 C1 3A 11 82 C9 79 D6 B9 C0
#A900 0E 20 C9 00 00 3C 01 4C 01 5A 01 00 18 10 C1 70
#A910 66 01 50 01 44 01 33 01 00 00 00 CD 80 A4 3A 27
#A920 82 A7 C0 0E 04 11 28 82 1A C6 60 6F 26 FF 06 0F
#A930 CD 2C A2 CD 88 A5 A7 CA 44 A9 05 C2 30 A9 13 0D
#A940 C2 28 A9 C9 11 20 00 19 7E D6 20 C0 22 26 82 36
#A950 D8 C9 3A 13 82 C6 41 6F 26 FF CD 88 A5 A7 C8 2B
#A960 2B CD 88 A5 A7 C8 3A 13 82 C3 AE A5 72 11 20 00
#A970 9C BE F2 FC 09 66 0F C3 9B BE 71 B4 4E 81 D4 C9
 #180 FF
]
OK
#
```

35 三次元効果を増す

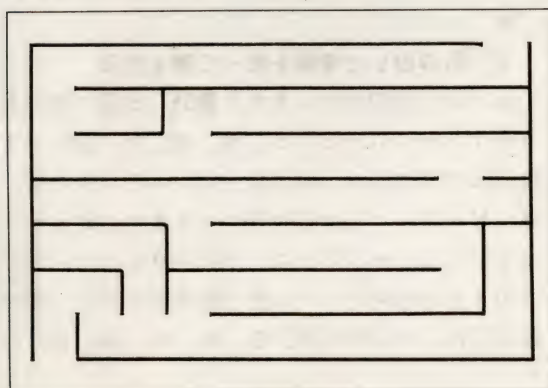
立体迷路プログラム

EX-80BS

吉沢　清

迷路と一口に言いましても色々ですが，一般的なものといえば**第35－1図**に示す様なものでよく目にされると思います。又この迷路をマイコンに作らせては，と考えられたものもよく本に掲載されている様です。しかしこれだけでは，あまりにも味けなさすぎます。そこでTV・DISPのグラフィックを利用し，あたかも自分が迷路の壁の中を歩いている様に作りましたのが，この立体迷路プログラムです。

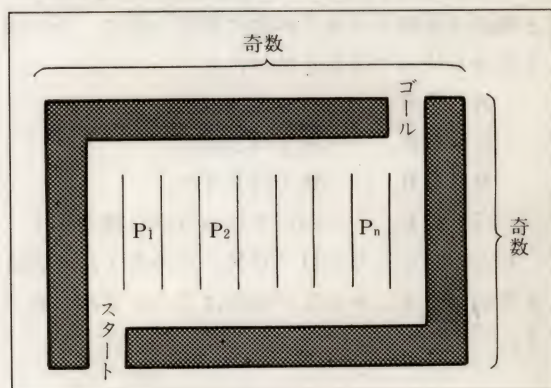
ＡＣＴ１迷路の作り方

①**第35－2図**の様に奇数×奇数のマスで枠を作り，偶数にあたる部分を２か所開けてスタートとゴールにします。又，縦，横が奇数にあたる部分をそれぞれP_1～P_nまでポイントとして番号をふります。

《第35－１図》

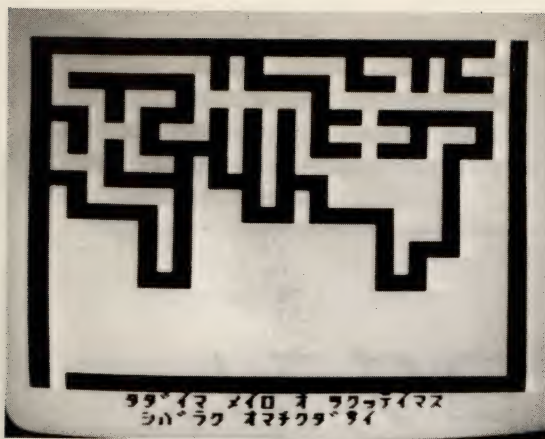
```
ゲーム ノ セツメイ ハ ヒツヨオデスカ ? Y

イマカラ メイロ オ ツクリマス
メイロ ガ デキルト ガメン ハ
メイロ ノ ナカ オ リッタイテキ ニ
ヒョウジ シマス

シタ ニ シメス メイレイ オ ツカッテ
メイロ オ ヌケダシテ クダサイ

R「CR」--ミギ オ ムキマス
L「CR」--ヒダリ オ ムキマス
B「CR」--ウシロ オ ムキマス
G「CR」--ムイテイル ホオコオ ニ ススミマス

ワカラナク ナッタラ D「CR」 オ
メイレイ スレバ メイロ オ ヒョウジ
シマス アナタ ノ バショ ハ
↑ デ シメサレタ トコロデス

ヨミオワリ マシタカ ?
```

《写真35－１》　ゲームの説明

②枠につながっている壁を固定壁，そうでない壁を中立壁と定義します。

③初期ポイントP_1から壁を作っていきます。まずP_1を基点としてランダムに求められた方向に向って二つ進み，そこに中立壁がなければ基点からそ

《第35－２図》

《写真35－2》 迷路を作成中

《写真35－3》 迷路が見えて来ました

の場所まで中立壁を作り，新しい基点とします。もし，求められた方向に中立壁があるときは，方向を90°変え同様に行ないます。又，360°方向を変えてもだめな場合（四方が中立壁に囲まれた場合）は，①に戻り最初から作り直します。

壁がどんどん延びていき固定壁にぶつかった時は，枠につながったものとして，最初のポイントから作ってきた壁をすべて固定壁にします。

④ポイントを次に移動して同様に壁を作っていきますが，もし，最初のポイントに前のポイントによって作られた固定壁があれば，そのポイントでは何もせず次に移ります。

⑤最後のポイントまで終ると迷路ができ上がります。

ACT 2 遊び方

プログラムをRUNしますと以下の様に説明してきます。

ゲームの説明は必要ですか？

[Y] [CR]（KEY－INする）

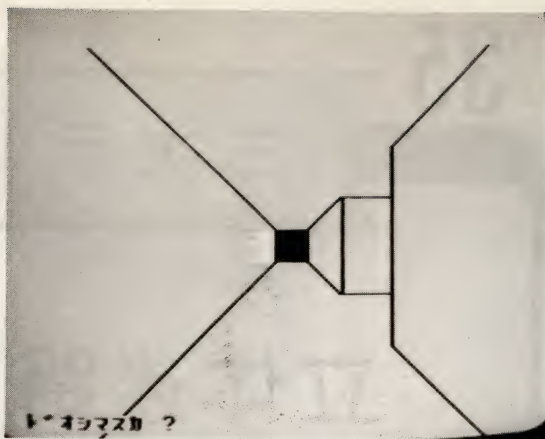
今から迷路を作ります。迷路ができ上がりますと画面は迷路の中を立体的に表示します。下に示す命令を使って迷路を抜け出して下さい。

R「CR」……右を向きます。

L「CR」……左を向きます。

B「CR」……後を向きます。

G「CR」……向いている方向に進みます。

判らなくなったらD「CR」を命令すれば迷路を表示します。あなたの場所は↑で示された所です。

読み終りましたか？

[Y] [CR]（KEY－INする）

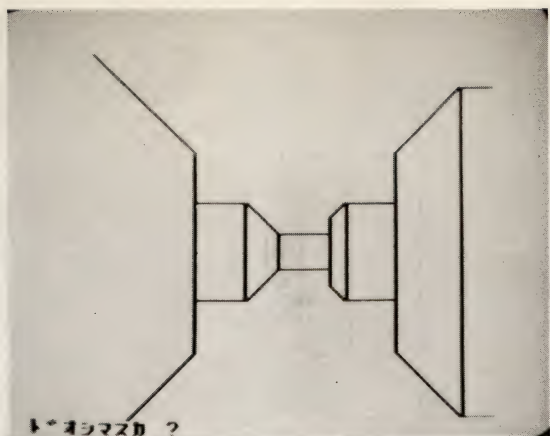
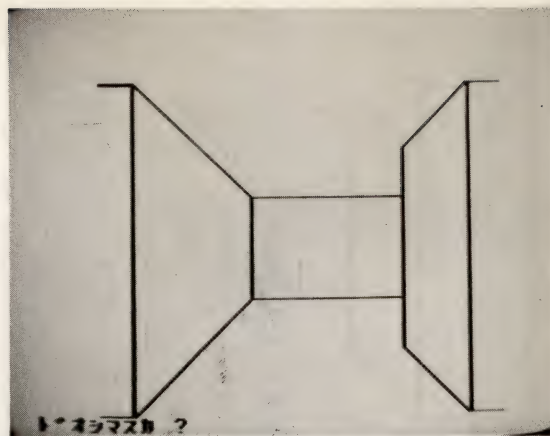
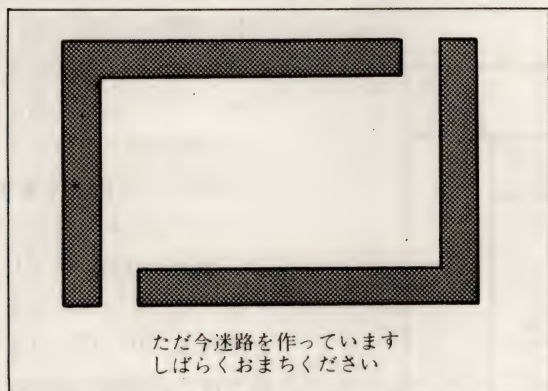
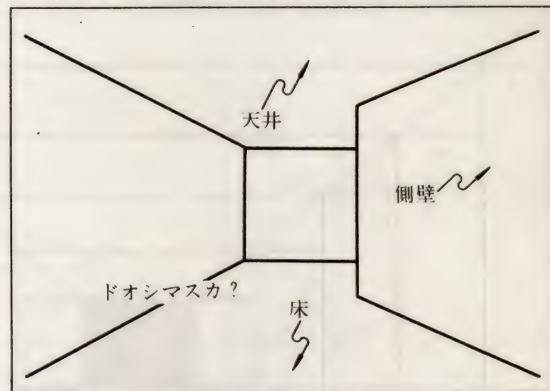
以上説明が終るとプログラムは初期設定され，迷路を作り始めます。まず枠が作られ，それが終ると中に一本一本壁を作っていきます（**第53－3図**）（途中、迷路作りに失敗しますと初期設定からやり直すことがあります）…迷路ができ上がるとスタートポイントに立たされ，そとから見た迷路の中を表示し，ドウシマスカ？と聞いてきます（**第35－4図**）。さて，ここから上の説明にあった様なコマンドを使用して迷路を無事，抜け出してください。ゴールに着くと，GAME・OVERと表示され，歩いてきた歩数と迷路を参照した回数が表示されます。

ACT 3 どのように立体図を書かせるか

1）迷路を読み取る

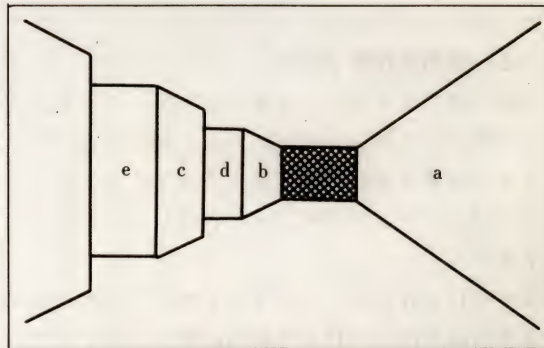
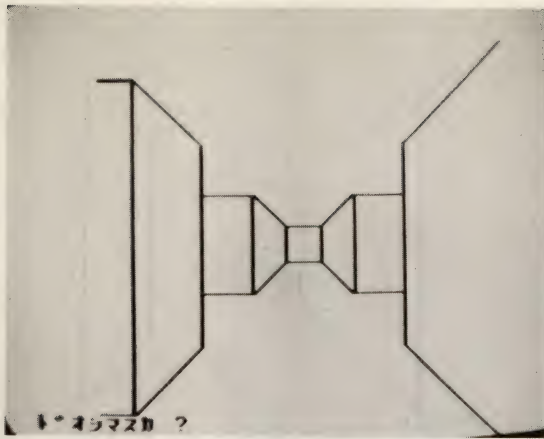
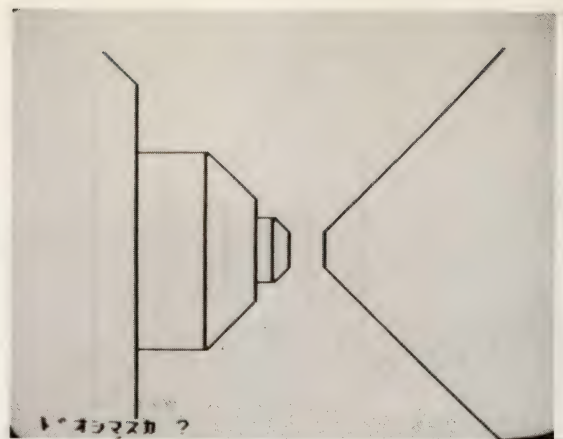
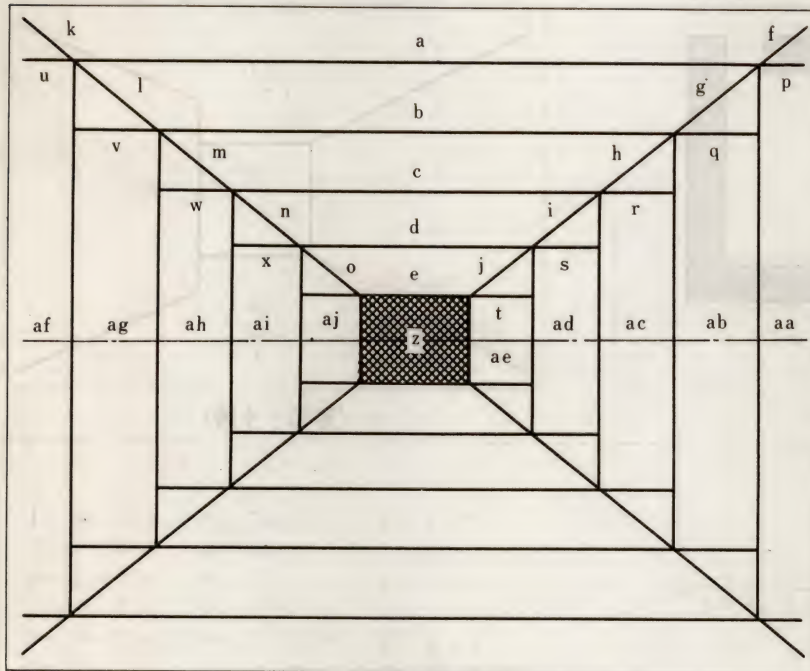
現在立っている位置から方向に向って一つ左から順に読み出し，@（U＋1）～@（U＋18）に入れる。（U＝795）壁は“1”壁のない所は“0”とする。

2）読み出した情報を使って壁を作る

(A)に示す情報を使いますと**第35－5図**の様になります。@（U＋3，6，9，12，15，18）＝1は，aの壁になり，@（U＋13）＝1はbの壁，@（U＋7）＝1はCの壁になります。dとeに当る部分は，@（U＋4，10）＝0なので，bとcのこちらを向いている壁が表示されます。壁の組立ては，@（U＋2，5，8，11，14，17）の

《写真35－4》 入りくんだ迷路です

《写真35－5》 ここでまがるか先に行くか？

《第35－3図》

《第35－4図》

A図

列を手前から見ていき，最初に“1”の立っている場所を，突き当たった壁としてセットし，以下手前に向って組立てます。なお，(A)の例の様に列全部がすべて“0”の場合，表示不能と言うことで一番奥を黒くぬりつぶしてしまいます。

これらの壁を線で表示するためには，**第35－6図**に示すだけの線を必要とします。どの様に線を選んでいくかはフローチャートを参考にしてください。又，ＢＳでは上半分しか線を引いていませ

《第35－5図》

《写真35－6》 まがったのはいいが……

《写真35－7》 出口が見えて来ました

《第35－6図》

んが，表示される迷路はすべて上下対象であるために，ＢＳが書き終えた後，機械語で鏡に写した様に書き込み，メモリ節約とスピードをはかるためです。

3）縦の線がポイント

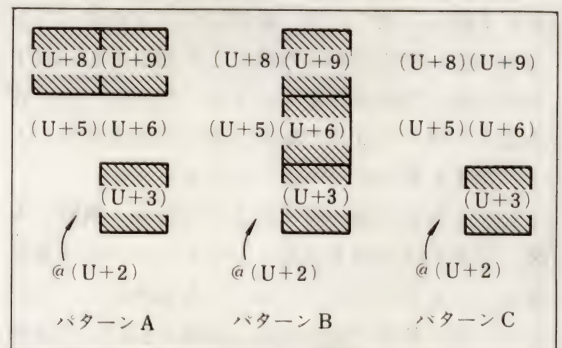
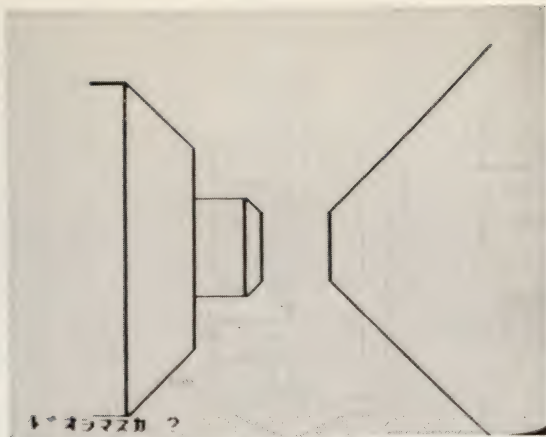
横，斜の線を引くには簡単な判定，たとえばそこに壁があるかないか，といった様な判定で引くことが出来ますが，縦はそううまくいきません，ではプログラムはどのように判定しているのでしょうか。

＠（Ｕ＋５，６，８，９）を使って説明します。立体図は常に＠（Ｕ＋２）の位置から見える壁によって描かれますので，縦の線を引かなくてよい条件は**第35－７図**の様になります。つまり図にある両パターン以外と判定した時縦の線を引けばよいと言うことになります。さて図ではプログラムになりませんので，＠（Ｕ＋５，６，８，９）をそれぞれＡ，Ｂ，Ｃ，Ｄと置き換えて論理式を作ってみます。

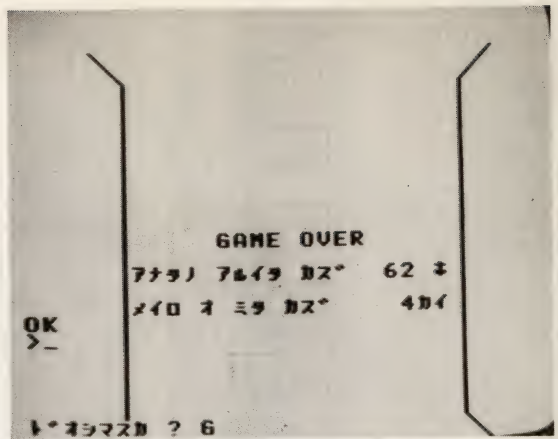
$\overline{A}\cdot\overline{B}\cdot C\cdot D\vee\overline{A}\cdot B\cdot\overline{C}\cdot D\vee\overline{A}\ \overline{B}\ \overline{C}\ \overline{D}=0$

上の式を満足する時線を引きます。

このプログラムは６Ｋバイトのメモリを必要とします。ＥＸ－８０ＢＳの場合ですと最終アドレスがＡ０００Ｈまで必要となります。

又，ＬＩＳＴを見ていただくとわかりますが，

《第35－７図》

《写真35－8》 出口が近づいて来ます

《写真35－9》 やっと終了です

1REMと2REMは変な記号とカタカナになっていますが，ここにはBSとリンケージを取ってある機械語が入っています。作り方は最初にREM文を書き文字のかわりにスペースをできるだけ入れます。次にモニタをAに移しスペースの入っている所に機械語を書込んでいきます。あとはBSの中でCALLするだけでOKです。この様にしますとテープからのプログラムのロードが一回ですみ大変便利です。

立体迷路作成G・F

START

(U+5)=1 — YES → aを作図(注1) → (U+5)(U+6) (U+2)(U+3) 〔aa〕(注2) → (U+3)=1 — YES → f ／ NO → (U+6)=1 — YES → p ／ NO → (U+5)(U+4) (U+2)(U+1) 〔af〕 → (U+1)=1 — YES → k ／ NO → (U+4)=1 — YES → u ／ NO

(U+8)=1 — YES → b → (U+8)(U+9) (U+5)(U+6) 〔ab〕 → (U+6)=1 — YES → g ／ NO → (U+9)=1 — YES → q ／ NO → (U+8)(U+7) (U+5)(U+4) 〔ag〕 → (U+4)=1 — YES → l ／ NO → (U+7)=1 — YES → v ／ NO

(U+11)=1 — YES → c → (U+11)(U+12) (U+8)(U+9) 〔ac〕 → (U+8)=1 — YES → h ／ NO → (U+12)=1 — YES → r ／ NO → (U+11)(U+10) (U+8)(U+7) 〔ah〕 → (U+7)=1 — YES → m ／ NO → (U+10)=1 — YES → w ／ NO

(U+14)=1 — YES → d → (U+14)(U+15) (U+11)(U+12) 〔ad〕 → (U+12)=1 — YES → i ／ NO → (U+15)=1 — YES → s ／ NO → (U+14)(U+13) (U+11)(U+10) 〔ai〕 → (U+10)=1 — YES → n ／ NO → (U+13)=1 — YES → x ／ NO

(U+17)=1 — YES → e → (U+17)(U+18) (U+14)(U+15) 〔ae〕 → (U+15)=1 — YES → j ／ NO → (U+18)=1 — YES → t ／ NO → (U+17)(U+16) (U+14)(U+13) 〔aj〕 → (U+13)=1 — YES → o ／ NO → (U+16)=1 — YES → y ／ NO

NO → 注3 — YES → Z(黒くぬりつぶされた面) ／ NO

END

注1：第35-6図で示す線を表示する。以下小文字だけで書いてあるフローも同様である。

注2：

C	D
A	B
〔X〕	

。 $\overline{A}\cdot\overline{B}\cdot C\cdot D\vee\overline{A}\cdot B\cdot\overline{C}\cdot D\vee\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}=0$ の条件を満たした時〔X〕で示される線を表示する。

注3：$(U+16)\vee(U+17)\vee(U+18)=0$か？

プログラムリスト

```
   1 REM     ﾅ1!ﾀﾊﾊﾊﾊｳﾊｶﾊ｢#ﾃ
   2 REMｯB17090 ｯB4ﾁ/>ﾃ>ﾃ>ﾃ>｡ﾃ>ｶ
ﾃ>ｳﾃ
   3 REM   *** ﾒｲﾛ ｹﾞｰﾑ ***
   4 CL.;GOS.7000
   5 F.I=1TO630;@(I)=0;N.I
   7 GOS.2050
   8 @(815)=0,@(816)=0
  10 F.I=1TO27;@(I)=1;N.I;@(29)=
1
  20 F.I=1TO20;@(I*30+1)=1;@(I*3
0+29)=1;N.I
  30 @(601)=1;F.I=3TO29;@(I+600)
=1;N.I
  35 CL.
  40 GOS.1000
  45 CU.5,22;P.'ﾀﾀﾞｲﾏ ﾒｲﾛ ｵ ﾂｸｯﾃ
ｲﾏｽ'
  46 P.'        ｼﾊﾞﾗｸ ｵﾏﾁｸﾀﾞｻｲ'
  50 F.I=1TO165
  60 C=I;GOS.2000;IFD=1G.270
 120 D=R.(4);O=0


 130 IF D=1E=S-60,P=C-15,K=S-30;
G.170
 140 IF D=2E=S+60,P=C+15,K=S+30;
G.170
 150 IF D=3E=S-2,P=C-1,K=S-1;G.1
70
 160 E=S+2,P=C+1,K=S+1
 170 IF@(630+P)=1G.240
 180 @(S)=1;@(630+C)=1
 190 @(K)=1;GOS.3000
```

```
 230 C=P;GOS.2000;IFD=1GOS.2050;
G.270
 235 G.120
 240 D=D+1,D=D-(D-1)/4*4
 250 O=O+1;IFO=4G.5
 260 G.130
 270 N.I;T=21,Y=2,H=0
 275 GOS.4000;GOS.500;CA.8435H
 280 L=320,R=330,G=350,B=340,D=3
90
 290 CU.0,23;IN.'ﾄﾞｵｼﾏｽｶ'A
 300 IF (A=L)+(A=R)+(A=G)+(A=B)+
(A=D) G.A


 310 G.290
 320 H=H+3,H=H-H/4*4;G.275
 330 H=H+1,H=H-H/4*4;G.275
 340 H=H+2,H=H-H/4*4;G.275
 350 IF H=0 X=T-1,R=Y
 360 IF H=1 R=Y+1,X=T
 370 IF H=2 X=T+1,R=Y
 380 IF H=3 R=Y-1,X=T
 382 IF X<1 G.8000
 383 S=(X-1)*30+R
 385 IF @(S)=1 CU.11,12;P.'ｶﾍﾞ ﾆ
ﾌﾞﾂｶﾘ ﾏｼﾀ';G.280
 387 IF X>=22 CU.10,12;P.'ｽﾀｰﾄ ﾎ
ﾟｲﾝﾄ ﾃﾞｽ';G.280
 388 T=X,Y=R,@(815)=@(815)+1;G.2
75
 390 CL.;GOS.1000
 395 @(816)=@(816)+1
 400 V=Y-1;CU.V,T
 410 IF H=0 P.'↑'
 420 IF H=1 P.'→'
 430 IF H=2 P.'↓'
```

```
 440 IF H=3 P.'←'
 450 G.280
 500 CL.;U=795;IF@(U+5)=0G.575
 505 F.I=6TO25;CU.I,2;P.' ';N.I
 510 M=@(U+2),N=@(U+3),O=@(U+5),
P=@(U+6),F=0
 515 GOS.6000;IFQ=0F.I=3TO12;CU.
26,I;P.'| ';N.I;F=1
 520 IF@(U+3)=1PO.0FCDAH,0B3H;PO
.0FCBBH,0B3H;G.540
 525 IF@(U+6)=0G.540
 530 IFF=0PO.0FCDAH,9DH;PO.0FCDB
H,9DH;G.540
 535 PO.0FCFAH,9EH;PO.0FCFBH,96H

 540 M=@(U+2),N=@(U+1),O=@(U+5),
P=@(U+4),F=0
 545 GOS.6000;IFQ=0F.I=3TO12;CU.
5,I;P.' |';N.I;F=1
 550 IF@(U+1)=1PO.0FCA4H,0B6H;PO
.0FCC5H,0B6H;R.
 555 IF@(U+4)=0R.


 560 PO.0FCC4H,9DH;PO.0FCC5H,9DH

 565 IFF=1PO.0FCE5H,9FH;PO.0FCE4
H,96H
 570 R.
 575 IF@(U+8)=0G.650
 580 CU.10,6;F.I=1TO12;P.'_',;N.
I
 585 M=@(U+5),N=@(U+6),O=@(U+8),
P=@(U+9),F=0
 590 GOS.6000;IFQ=0F.I=7TO12;CU.
22,I;P.'| ';N.I;F=1
 595 IF@(U+6)=1V=6;F.I=22TO25;CU
.I,V;P.'/';V=V-1;N.I;G.615
 600 IF@(U+9)=0G.615
 605 IFF=1PO.0FD76H,9EH;F.I=23TO
25;CU.I,7;P.'‾';N.I;G.615
 610 F.I=22TO25;CU.I,6;P.'_';N.I

 615 M=@(U+5),N=@(U+4),O=@(U+8),
P=@(U+7),F=0
 620 GOS.6000;IFQ=0F.I=7TO12;CU.
9,I;P.' |';N.I;F=1


 625 IF@(U+4)=1V=3;F.I=6TO9;CU.I
,V;P.'\';V=V+1;N.I;G.510
 630 IF@(U+7)=0G.510
 635 IFF=1PO.0FD69H,9FH;F.I=6TO8
;CU.I,7;P.'‾';N.I;G.645
 640 F.I=6TO9;CU.I,6;P.'_';N.I
 645 G.510
 650 IF@(U+11)=0G.720
 655 F.I=13TO18;CU.I,9;P.'_';N.I

 660 M=@(U+8),N=@(U+9),O=@(U+11)
,P=@(U+12),F=0
 665 GOS.6000;IFQ=0F.I=10TO12;CU
.19,I;P.'| ';N.I;F=1
 670 IF@(U+9)=1V=9;F.I=19TO21;CU
.I,V;P.'/';V=V-1;N.I;G.690
 675 IF@(U+12)=0G.690
 680 IFF=1PO.0FDD3H,9EH;PO.0FDD4
H,96H;PO.0FDD5H,96H;G.690
 685 F.I=19TO21;CU.I,9;P.'_';N.I

 690 M=@(U+8),N=@(U+7),O=@(U+11)
,P=@(U+10),F=0
 695 GOS.6000;IFQ=0F.I=10TO12;CU
.12,I;P.' |';N.I;F=1
 700 IF@(U+7)=1V=7;F.I=10TO12;CU
.I,V;P.'\';V=V+1;N.I;G.585
 705 IF@(U+10)=0G.585
 710 IFF=1PO.0FDCCH,9FH;PO.0FDCB
H,96H;PO.0FDCAH,96H;G.585
 715 F.I=10TO12;CU.I,9;P.'_';N.I
;G.585
 720 IF@(U+14)=0G.790
 725 F.I=14TO17;CU.I,10;P.'_';N.
I
 730 M=@(U+11),N=@(U+12),O=@(U+1
4),P=@(U+15),F=0
 735 GOS.6000;IFQ=0F.I=11TO12;CU
.18,I;P.'| ';N.I;F=1
 740 IF@(U+12)=1PO.0FDD2H,0B3H;G
.760
 745 IF@(U+15)=0G.760
 750 IFF=1PO.0FDF2H,9EH;G.760
 755 PO.0FDD2H,9DH
 760 M=@(U+11),N=@(U+10),O=@(U+1
4),P=@(U+13),F=0


 765 GOS.6000;IFQ=0F.I=11TO12;CU
.13,I;P.' |';N.I;F=1
 770 IF@(U+10)=1PO.0FDCDH,0B6H;G
.660
 775 IF@(U+13)=0G.660
 780 IFF=1PO.0FDEDH,9FH;G.660
 785 PO.0FDCDH,9DH;G.660
 790 IF@(U+17)+@(U+16)+@(U+18)=0
G.805
 793 IF@(U+17)=0G.870
 795 PO.0FDEFH,9DH;PO.0FDF0H,9DH

 805 M=@(U+14),N=@(U+15),O=@(U+1
7),P=@(U+18),F=0
 810 GOS.6000;IFQ=0PO.0FE11H,88H
;F=1
 815 IF@(U+15)=1PO.0FDF1H,0B3H;G
.835
 820 IF@(U+18)=0G.835
 825 IFF=1PO.0FE11H,9EH;G.835
 830 PO.0FDF1H,9DH
 835 M=@(U+14),N=@(U+13),O=@(U+1
7),P=@(U+16),F=0


 840 GOS.6000;IFQ=0PO.0FE0EH,8FH
;F=1
 845 IF@(U+13)=1PO.0FDEEH,0B6H;G
.730
 850 IF@(U+16)=0G.730
 855 IFF=1PO.0FE0EH,9FH;G.730
 860 PO.0FDEEH,9DH;G.730
 870 PO.0FE0FH,80H;PO.0FE10H,80H
;G.805
1000 N=0,M=1,O=1
1010 F.I=1TO21;CU.N,M
1020 F.J=1TO30
1030 B=@(O)
1040 IFB=1P.'▒',;G.1060
1050 P.' ',
1060 O=O+1;N.J
1070 N=0,M=M+1
1080 N.I;R.
2000 S=(C-1)/15*60+(C-(C-1)/15*1
5)*2-1
2010 D=@(S);R.
2050 F.J=1TO165;@(630+J)=0;N.J;R
.

3000 L=(S-1)/30+1,M=S-(S-1)/30*3
0-1
3010 CU.M,L;P.'▒',
3020 L=(K-1)/30+1,M=K-(K-1)/30*3
0-1
3030 CU.M,L;P.'▒',;R.
4000 Z=1,V=T,W=Y
4005 IF H=0 W=W-1
4010 IF H=1 V=V-1
4020 IF H=2 W=W+1
4030 IF H=3 V=V+1
4040 F.I=1TO6
4045 F.J=1TO3
4050 GOS.5000;Z=Z+1
4060 IF H=0 W=W+1
4070 IF H=1 V=V+1
4080 IF H=2 W=W-1
4090 IF H=3 V=V-1
4100 N.J
4110 IF H=0 V=V-1,W=W-3
4120 IF H=1 W=W+1,V=V-3
4130 IF H=2 V=V+1,W=W+3
```

```
4140 IF H=3 W=W-1,V=V+3
4150 N.I;R.
5000 S=(V-1)*30+W
5010 IF (V<1)+(V>22)+(W<1)+(W>
)G.5040
5030 @(795+Z)=@(S);R.
5040 @(795+Z)=0
6000 IFO*P*(M=0)*(N=0)+N*P*(M=0)
*(O=0)Q=1;R.
6001 IFM+N+O+P=0Q=1;R.
6002 Q=0;R.
7000 Y=7010;IN.'ｹﾞｰﾑ ﾉ ｾﾂﾒｲ ﾊ ﾋﾂ
ﾖｵﾃﾞｽｶ'A
7005 IFA=YG.A
7006 R.
7010 P.;P.'ｲﾏｶﾗ ﾒｲﾛ ｵ ﾂｸﾘﾏｽ'
7020 P.'ﾒｲﾛ ｶﾞ ﾃﾞｷﾙﾄ ｶﾞﾒﾝ ﾊ'
7030 P.'ﾒｲﾛ ﾉ ﾅｶ ｵ ﾘｯﾀｲﾃｷ ﾆ'
7040 P.'ﾋｮｳｼﾞ ｼﾏｽ';P.
7050 P.'ｼﾀ ﾆ ｼﾒｽ ﾒｲﾚｲ ｵ ﾂｶｯﾃ'
7060 P.'ﾒｲﾛ ｵ ﾇｹﾀﾞｼﾃ ｸﾀﾞｻｲ'
7070 P.;P.'R「CR」・・ﾐｷﾞ ｵ ﾑｷﾏｽ'
7080 P.'L「CR」・・ﾋﾀﾞﾘ ｵ ﾑｷﾏｽ'
7090 P.'B「CR」・・ｳｼﾛ ｵ ﾑｷﾏｽ'
7100 P.'G「CR」・・ﾑｲﾃｲﾙ ﾎｵｺｵ ﾆ ｽｽﾐﾏ
ｽ'
7110 P.;P.'ﾜｶﾗﾅｸ ﾅｯﾀﾗ D「CR」 ｵ'
7120 P.'ﾒｲﾚｲ ｽﾚﾊﾞ ﾒｲﾛ ｵ ﾋｮｳｼﾞ'
7130 P.'ｼﾏｽ ｱﾅﾀ ﾉ ﾊﾞｼｮ ﾊ'
7140 P.'↑','ﾃﾞ ｼﾒｻﾚﾀ ﾄｺﾛﾃﾞｽ'
7150 P.;IN.'ﾖﾐｵﾜﾘ ﾏｼﾀｶ'A
7160 IFA=YR.
7170 G.7150
8000 CU.11,12;P.'GAME OVER'
8005 A=@(815),B=@(816)
8010 CU.6,14;P.'ｱﾅﾀﾉ ｱﾙｲﾀ ｶｽﾞ',#
4,A,' ﾎ'
8020 CU.6,16;P.'ﾒｲﾛ ｵ ﾐﾀ ｶｽﾞ ',#
4,B,'ｶｲ';S.
OK
>_
```

機 械 語

*1␣REM

アドレス	機械語	ラベル	ニモニック
8435	F5		PUSH PSW
36	C5		PUSH B
37	D5		PUSH D
38	E5		PUSH H
39	2180FF		LXIH, FF80H
3C	11A0FC		LXID, FCA0H
3F	01C0FF		LXIB, FFE0H
42	1A	L8	LDAX D
43	FE96		CPI 96H
45	CA8984		JZ L1
48	FE9D		CPI 9DH
4A	CA8E84		JZ L2
4D	FE9E		CPI 9EH
4F	CA9384		JZ L3
52	FE9F		CPI 9FH
54	CA9884		JZ L4
57	FEB3		CPI B3H
59	CA9D84		JZ L5

アドレス	機械語	ラベル	ニモニック
845C	FEB6		CPI B6H
5E	CAA284		JZ L6
61	77	L9	MOV M, A
62	23		INX H
63	13		INX D
64	7D		MOV A, L
65	E61F		ANI IFH
67	C37484		JMP L7

*2␣REM

アドレス	機械語	ラベル	ニモニック
8474	C24284	L7	JNZ L8
77	09		DAD B
78	7D		MOV A, L
79	FE00		CPI 00H
7B	C24284		JNZ L8
7E	7C		MOV A, H
7F	FEFE		CPI FEH
81	C24284		JNZ L8
84	E1		POP H
85	D1		POP D
86	C1		POP B
87	F1		POP PSW
88	C9		RET
89	3E9D	L1	MVI A, 9DH
8B	C36184		JMP L9
8E	3E96	L2	MVI A, 96H
90	C36184		JMP L9
93	3EA0	L3	MVI A, A0H
95	C36184		JMP L9
98	3EA1	L4	MVI A, AIH
9A	C36184		JMP L9
9D	3EB6	L5	MVI A, B6H
9F	C36184		JMP L9
A2	3EB3	L6	MVI A, B3H
A4	C36184		JMP L9

36 リアルタイムゲーム3題

Ⅰ.もぐらたたき……………………前田茂穂

Ⅱ.ブロックくずし………………前田茂穂

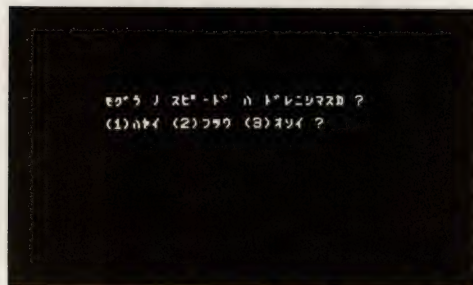
Ⅲ.早打ちマック…………………和田次雄

MZ-80K

●もぐらたたきゲーム●

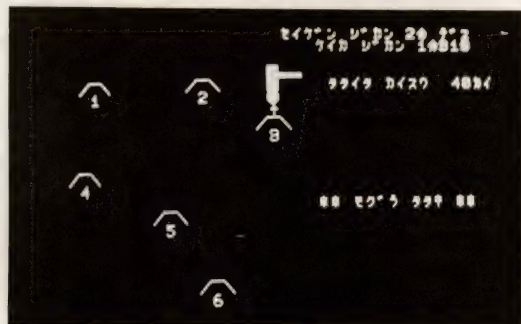
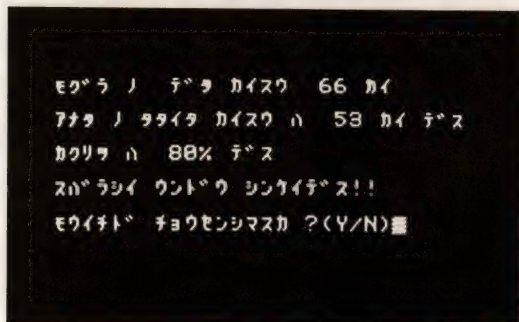
このゲームは，街のゲームコーナーなどにあるもぐらたたきのゲームをＴＶ画面用にアレンジしたものです。

画面上の山のでてくるモグラの数字のキーをうまくおせば，上からトンカチでたたいて得点します。

プログラムを入力して，〔RUN〕するとモグラの出現するスピードを入力してください。(1)ハヤイにしますと，本当にはやいですから注意してください。制限時間の２分間に何匹のモグラをたたいたかを競います。モグラをたたいた数によりゲーム終了時に，モグラの出現した回数とたたいた回数，確率が表示され，それに合ったコメントが出ます。ついついあせってキーボードを力いっぱいたたいてしまうゲームです。キー・ボードを大切にして，ゲームを楽しんで下さい。

《写真36－１》最初スピードの設定

《写真36－２》ゲームスタート

《写真36－３》また一匹アウト

モク゛ラ ノ デ゛タ カイスウ 66 カイ
アナタ ノ タタイタ カイスウ ハ 53 カイ デ゛ス
カクリツ ハ 80% デ゛ス
スハ゛ラシイ ウント゛ウ シンケイデ゛ス!!
モウイチト゛ チョウセンシマスカ ?(Y/N)

《写真36－４》ゲーム終了時の表示

もぐらたたきゲーム・プログラム・リスト

```
1 REM"ﾓｸﾞﾗ ﾀﾀｷ"
5 A0=83:A1=93:A2=219:A3=362:A4=490:A5=694:A6=1:A7=207:A8=80:A9=23:Q=0:M=0
6 P=53248:V=40:TEMPO7
7 PRINT"■■■■■■■■■■■■ﾓｸﾞﾗ ﾉ ｽﾋﾟｰﾄﾞ ﾊ ﾄﾞﾚﾆｼﾏｽｶ ?"
8 PRINT"■■■■(1)ﾊﾔｲ (2)ﾌﾂｳ (3)ｵｿｲ ";:INPUT X:ONX GOTO600,620,640:GOTO7
10 TI$="000000":PRINT"■";
15 PRINT"■■■■■■/‾＼";"■";
16 PRINT"■■■■■■■■■■■■■■■■■/‾＼"
17 PRINT"■■■■■■■■■■■■■■■■■■■■■■■/‾＼"
18 PRINT"■■■/‾＼"
19 PRINT"■■■■■■■■■■■/‾＼"
20 PRINT"■■■■■■■■■■■■■■■■■■■/‾＼"
21 PRINT"■";
22 PRINT"■■■■■■■■1";"■";
23 PRINT"■■■■■■■■■■■■■■■■■■■■■2"
24 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■3"
25 PRINT"■■■■4"
26 PRINT"■■■■■■■■■■■■■■5"
27 PRINT"■■■■■■■■■■■■■■■■■■■■■6"
28 PRINT"■";TAB(20);"ｾｲｹﾞﾝ ｼﾞｶﾝ 2分 ﾃﾞｽ"
29 PRINT"■■■■■■■■■■■■■■■■";TAB(23);"** ﾓｸﾞﾗ ﾀﾀｷ **":Q=0
30 A=0
40 GOSUB700
43 B=INT(RND(1)*7):ONBGOTO50,60,70,80,90,100
45 GOTO40
50 C=A0:GOTO110
60 C=A1:GOTO110
70 C=A2:GOTO110
80 C=A3:GOTO110
90 C=A4:GOTO110
100 C=A5
110 Q=Q+A6:POKEP+C,A7:POKEP+C+V,A8:MUSIC"‾A0":FORQE=1TO X
121 IFVAL(MID$(TI$,4,1))>=2 THEN1000
128 GETD:IFB=DTHEN200
130 NEXT QE:GOTO 320
140 GOTO120
200 IFA>1THENPOKEP+C,0:POKEP+C+V,0:GOTO30
205 POKEP+C-V,67:POKEP+C-V*2,67:POKEP+C-V*2+1,126:POKEP+C-V*2+2,126
210 GOTO500
320 POKEP+C,0:POKEP+C+V,0:GOTO30
400 FORJ=1TO30:NEXTJ:RETURN
500 FORI=1TO5:POKEP+C,206:GOSUB400:MUSIC"‾G0":GOSUB400:POKEP+C,207:NEXTI
505 POKEP+C-V,0:POKEP+C-V*2,0:POKEP+C-V*2+1,0:POKEP+C-V*2+2,0
510 M=M+1:PRINT"■■■■■";TAB(24);"ﾀﾀｲﾀ ｶｲｽｳ ";M;"ｶｲ":GOTO320
600 X=10:GOTO10
620 X=30:GOTO10
640 X=50:GOTO10
700 PRINT"■";"■";TAB(A9);"ｹｲｶ ｼﾞｶﾝ ";MID$(TI$,4,1);"分";MID$(TI$,5,2);"秒"
710 RETURN
1000 PRINT"■■■■■■■■■■";TAB(7);"ﾓｸﾞﾗ ﾉ  ﾃﾞﾀ ｶｲｽｳ ";Q;" ｶｲ":PRINT
1010 PRINTTAB(7);"ｱﾅﾀ ﾉ ﾀﾀｲﾀ ｶｲｽｳ ﾊ ";M;" ｶｲ ﾃﾞｽ":PRINT
1015 W=INT((M/Q)*100)
1020 PRINTTAB(7);"ｶｸﾘﾂ ﾊ ";W;"% ﾃﾞｽ"
1030 IFW<50THEN1100
1040 IFW<80THEN1200
1050 PRINT:PRINTTAB(7);"ｽﾊﾞﾗｼｲ ｳﾝﾄﾞｳ ｼﾝｹｲﾃﾞｽ!!"
1060 PRINT:PRINTTAB(7);:INPUT"ﾓｳｲﾁﾄﾞ ﾁｮｳｾﾝｼﾏｽｶ ?(Y/N)";X$
1070 IFX$="Y"THEN 5
1080 END
1100 PRINT:PRINTTAB(7);"ｽｺｼ ﾆﾌﾞｲﾃﾞｽﾈ!!"
1110 GOTO1060
1200 PRINT:PRINTTAB(7);"ﾋﾄﾅﾐ ﾉ ｳﾝﾄﾞｳ ｼﾝｹｲﾃﾞｽ!!"
1220 GOTO1060
```

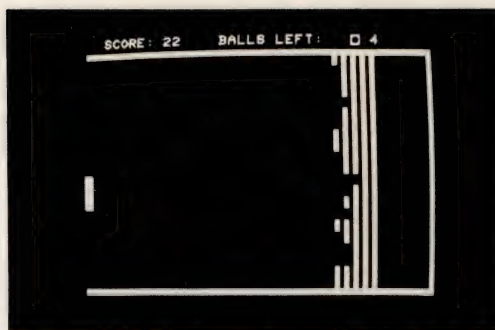
●ブロックくずし●

ゲーム自体はとても有名なものなので，いまさらルールの説明はいらないでしょう。ただテレビゲームは，画面の上の方にブロックがありますが，このゲームでは画面を横に使います。

プログラムは，全てBASICで組んでありますが，画面表示をPOKE文で制御していますので，ゲーム進行はスムーズに行なえます。

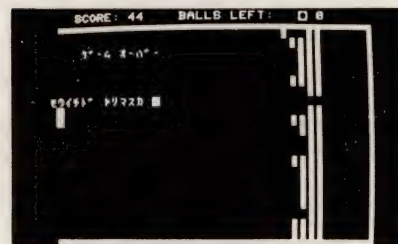
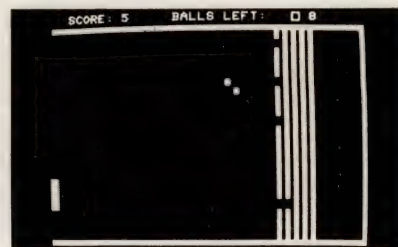
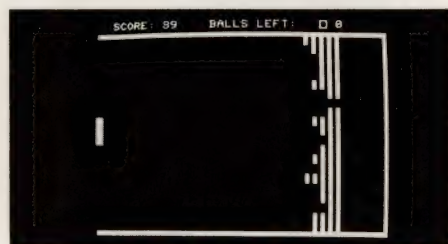
まずプログラムを入れたら〔RUN〕してください。

最初に使用するボールの数を聞いて来ますのでその個数を入力すればゲームはスタートです。

①キーで上，②キーでストップ，③のキーで下に画面上のラケットが移動します。

ブロックをこわして，中に侵入した場合はラケットの大きさが小さくなります。

⑦キーでボールをサーブします。

《写真36－５》ブロックくずし画面表示

ブロックくずし画面表示

ブロックくずしプログラムリスト

```
110 PRINT"■■■";TAB(7);"┌──────────────────────┐"
120 PRINTTAB(7);"│    BLOCK KUZUSHI GAME    │"
130 PRINTTAB(7);"└──────────────────────┘"
140 PRINT"■■■■■";:INPUT"ﾎﾞｰﾙ ﾉ ｶｽﾞ ﾊ?";P1:IFP1<1THEN140
150 T=0:H=0:L=1000:D=53248:M=D+81:G1=123:G2=59:G3=71:G4=90:G5=0:N=40
152 W=0:W1=26:W2=32:W3=1:W4=3:W5=-1:W6=35:W7=20:W8=105:W9=18:WW=2
200 PRINT"■■■":INPUT"ｱﾅﾀ ﾉ ﾅﾏｴ ﾊ ?";N$:N$=LEFT$(N$,9)
210 PRINT"■   SCORE: 0      BALLS LEFT:";P1-1
220 PRINT"  ══════════════════════════════":GOSUB1010
230 FORI=0TO20:POKED+118+N*I,G2:NEXT
240 PRINT"■";:FORQ=1TO23:PRINT"■";:NEXT
245 PRINT"  ══════════════════════════════";
250 IFT<>0THENPRINT:PRINT"HIGH:";H;"(";H$;")";TAB(20);"LOW:";L;"(";L$;")";
260 FORI=1TO3:POKED+N*(9+I)+1,G4:NEXT
270 S=0:P=P1-1:R1=1:R=9
310 X=0:Y=INT(RND(1)*16+2):A=1:B=SGN(RND(1)-.5):U=1:V=1
```

```
320 C=M+W3+X+N*Y:F=PEEK(C):POKEC,G3
330 IF(X>=W)*(X<W1)+(X>W2)THEN510
410 O=C+U:K=N*V:TT=C+K
415 IFF=G1THENGOSUB2005:IFPEEK(O+K)=G1THENA=-A:B=-B:GOTO510
420 IFPEEK(O)=G1THENGOSUB2005:POKEO,G5:IFPEEK(C+WW*U)=G1THENA=-A:GOTO510
430 IFPEEK(TT)=G1THENGOSUB2005:POKETT,G5:IFPEEK(TT)=G1THENB=-B
510 GETE:IF(E<W3)+(E>W4)THENE=G
520 ONEGOSUB710,910,810
530 G=E
600 POKEC,G5
610 IF(X=W)*(U=W5)THENONR1GOTO3010,4005
620 IFX=W6THENA=-A:MUSIC"‾F0":IFR1=W3THENR1=2:POKEM+(R+W3)*N,G5
630 IF(Y=W)+(Y=W7)THENB=-B:MUSIC"‾F0"
640 X=X+A:Y=Y+B:U=SGN(A):V=SGN(B)
650 IF(Y<W)+(Y>W7)THENY=Y-V
670 IF(S=INT(S/W8)*W8)*(S<>W)*(X=W)THENGOSUB1010
680 GOTO320
710 IFR=W9+R1THENE=2:RETURN
720 R=R+W3:POKEM+N*(R+WW-R1),G4:POKEM+N*(R-WW),G5
730 RETURN
810 IFR=W3THENE=2:RETURN
820 R=R-W3:POKEM+N*(R+W4-R1),G5:POKEM+(R-W3)*N,G4
830 RETURN
910 RETURN
1010 PRINT"■■":FORI=0TO20
1020 PRINTTAB(29);"■■■■■":NEXT
1030 RETURN
2005 TEMPO7:MUSIC"G0"
2010 S=S+1:PRINT"■";TAB(9);S
2020 RETURN
3010 IFABS(R-Y)>=WWTHEN5010
3020 A=1:IFR-Y=W3THENB=INT(RND(1)*2)-2:GOTO4030
3030 IFR-Y=WTHENB=INT(RND(1)*3)-1:GOTO4030
3040 B=INT(RND(1)*2)+1:GOTO4030
4005 A=1:IFABS(R-Y)>=2THEN5010
4010 IFR-Y=WTHENB=INT(RND(1)*2)+1:GOTO4030
4020 IFR-Y=W3THENB=INT(RND(1)*2)-2:GOTO4030
4030 MUSIC"‾F0":GOTO620
5010 IFP=0THEN5050
5020 P=P-1:PRINT"■";TAB(27);"   ";:PRINT"□";TAB(27);P
5030 GETE:IFE<>7THEN5030
5040 R1=1:IFR=20THENR=19:POKEM+(R-1)*N,G4:GOTO310
5045 POKEM+(R+1)*N,G4:GOTO310
5050 IFS>HTHENH=S:H$=N$
5060 IFS<=LTHENL=S:L$=N$
5070 PRINT"■■■■■■■■■ｹﾞｰﾑ ｵｰﾊﾞｰ":PRINT"■■■"
5080 T=T+1:FORI=1TO 300:NEXT:INPUT"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ ";A$
5090 IFASC(LEFT$(A$,1))<>ASC("N")THEN200
5100 PRINT"■"
5110 END
```

●早打ちマック●

このゲームは，画面上のⓐマークを制限時間内にいかに多く打ち落とすかというゲームです。制限時間は 3 分間。＊マークが障害物としてたくさん画面上に現われています。

プログラムを入力して〔RUN〕するとゲームがスタートです。ランダムな位置に人間が登上します。

1のキー………人間左移動

2のキー………人間右移動

3のキー………人間上移動

4のキー………人間下移動

6のキー………弾左に発射

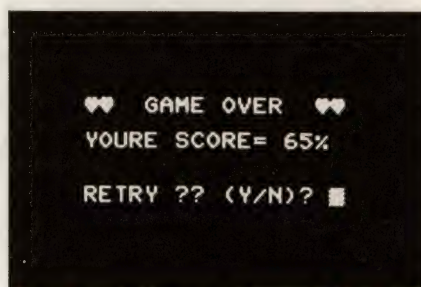
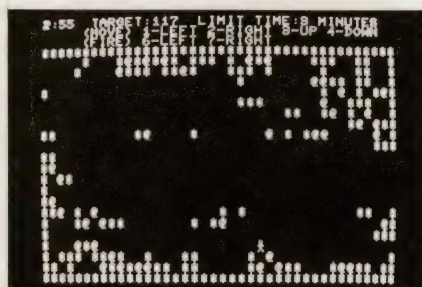
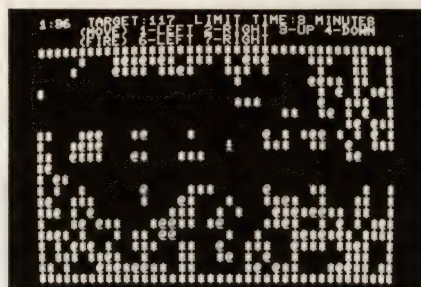
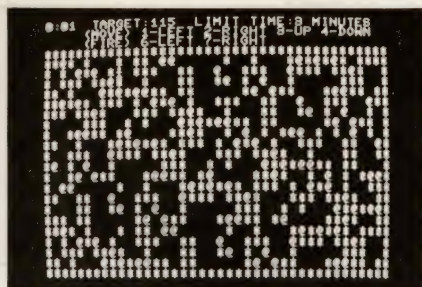
7のキー………弾右に発射

画面左上にタイマー機能をつけてあり，時間経過がわかります。あたえられたターゲットを 3 分間でいかに多く打ちたおすかを競うゲームです。

ゲーム終了時に打ちたおしたターゲットのパーセントがスコアとして表示されます。

弾の発射は，瞬間的に●マークを画面に表示して，さも弾を発射したように見せています。

この方法は，弾をあつかうゲームなどではスピード感を出すのに効果的な方法でしょう。

早打ちマック画面表示

早打ちマックプログラム・リスト

```
10 REM MACK
12 D=53248:D1=202:TEMPO7
13 TN=-1:T0=0:T1=1:T2=2:T3=3:T4=4:T5=5
14 TA=23:TB=24:TC=39:TD=40
16 DIM X(8),Y(8),M(10)
20 GOSUB 1000
30 GOSUB 1200
80 GOSUB 1100
90 TI$="000000"
92 REM
94 REM ***  MOVE  ***
96 REM
98 PRINT"▒▒    TARGET:";RIGHT$(STR$(Z),3);"  LIMIT TIME:3 MINUTES"
100 PRINT"▒";MID$(TI$,T4,T1);":";RIGHT$(TI$,T2)
105 IFVAL(MID$(TI$,T4,T1))>=T3THEN 2000
107 GET A
110 ON A GOTO 130,160,200,180,100,290,500
120 GOTO 100
130 IF PEEK(D+Y*TD+X-T1)<>T0THEN 100
140 POKE D+TD*Y+X,SP:X=X-T1
150 IF X<T0THEN X=T0
155 POKE D+TD*Y+X,D1:GOTO 100
160 IF PEEK(D+Y*TD+X+T1)<>T0THEN 100
170 POKE D+TD*Y+X,SP:X=X+T1
175 IF X>TD THEN X=TD :GOTO 155
176 GOTO 155
180 IF PEEK(D+(Y+T1)*TD+X)<>T0THEN 100
190 POKE D+TD*Y+X,SP:Y=Y+T1
193 IF Y>TB THEN Y=TB
196 GOTO 155
200 IF PEEK(D+(Y-T1)*TD+X)<>T0THEN 100
210 POKE D+TD*Y+X,SP:Y=Y-T1
213 IF Y<T0THEN Y=T0
216 GOTO 155
280 REM
283 REM ***  FIRE  ***
286 REM
290 IF X-T1<T0THEN 100
300 M0=PEEK(D+TD*Y+X-T1):POKED+TD*Y+X-T1,71:MUSIC"‾A0":POKED+TD*Y+X-T1,M0
310 I=T1
320 IF PEEK(D+TD*Y+X-I)<>T0THEN 370
330 I=I+T1:IF X-I<T0THEN 100
340 GOTO 320
370 MT=PEEK(D+TD*Y+X-I)
380 IF MT=ST THEN POKE D+TD*Y+X-I,SP:GOTO 100
390 X0=X-I:GOSUB 1500
400 IF Z=<T0THEN 2000
430 IF N<T0THEN 100
```

```
440 GOSUB1200:MUSIC"‾F9F‾FF‾FF‾FF":GOSUB1120:GOTO100
494 REM
495 REM ##########
496 REM
500 IF X+T1>TC THEN 100
510 M0=PEEK(D+TD*Y+X+T1):POKED+TD*Y+X+T1,71:MUSIC"‾A0":POKED+TD*Y+X+T1,M0
520 I=T1
530 IF PEEK(D+TD*Y+X+I)<>T0THEN 560
540 I=I+T1:IF X+I>TC THEN 100
550 GOTO 530
560 MT=PEEK(D+TD*Y+X+I)
570 IF MT=ST THEN POKE D+TD*Y+X+I,SP:GOTO100
580 X0=X+I:GOSUB 1500
590 IF Z=<T0THEN 2000
620 IF N<T0THEN 100
630 GOSUB1200:MUSIC"‾F9F‾FF‾FF‾FF":GOSUB1120:GOTO100
990 REM
992 REM ***  MAP CORDINATE  ***
994 REM
1000 ST=107:PH=85:SP=0:Z=0:C1=10
1010 PRINT"■"
1015 PRINT"     (MOVE) 1-LEFT 2-RIGHT 3-UP 4-DOWN"
1016 PRINT"     (FIRE) 6-LEFT 7-RIGHT"
1020 Y=3:FOR X=0TO39:POKED+TD*Y+X,ST:NEXT
1030 X=39:FORY=4TO24:POKED+TD*Y+X,ST:NEXT
1040 Y=24:FORX=39TO0STEP-1:POKED+TD*Y+X,ST:NEXT
1050 X=0:FORY=24TO4STEP-1:POKED+TD*Y+X,ST:NEXT
1057 FOR C=T0 TO T4
1058 RESTORE:C1=INT(RND(1)*TD)
1059 FOR B=T1 TO C1:READ MM:NEXT B
1060 FORY=T4+T4*CTOT4+T4*C+T3:FORX=T1TOTC-T1
1070 READ MM
1075 IF MM=PH THEN Z=Z+T1
1080 POKE D+TD*Y+X,MM
1085 NEXT X,Y,C:Z0=Z:RETURN
1090 DATA 107,85,0,107,85,107,0,107,0,0,0,107,0,107,107,0,85,85,0,0
1091 DATA 0,0,0,107,0,0,107,0,0,0,107,85,107,85,107,85,107,0,107,107
1092 DATA 107,0,107,0,85,107,85,0,107,0,0,107,107,107,0,0,107,107,0,0
1093 DATA 85,0,0,107,107,107,0,0,85,107,107,107,0,107,107,85,107,0,0,0
1094 DATA 107,107,107,0,107,107,85,107,0,0,107,0,0,0,107,107,0,0,107,85
1095 DATA 0,107,0,85,0,0,107,0,0,0,0,0,107,107,107,0,0,107,0,107
1096 DATA 0,0,0,85,107,85,107,85,85,107,107,0,107,107,0,0,0,0,0,107
1097 DATA 85,0,85,107,107,0,0,0,107,85,85,107,107,0,107,85,0,0,0,0
1098 DATA 107,107,85,107,0,0,107,0,85,0,107,0,107,85,85,0,107,0,107,0
1099 DATA 85,0,107,0,107,0,107,0,0,107,0,107,107,0,0,10,0,85,107,107,0,0,0
1100 REM
1101 REM  ***  KAITEN  ***
1102 REM
1110 REM VAR;X,Y
1120 D1=202
1125 FOR J=T1TOT5
1130 FOR I=T3 TOT0 STEP TN
1140 POKE D+TD*Y+X,D1+I:MUSIC"A1‾A"
1150 NEXT I,J
1160 RETURN
1200 REM MAN'S POSITION
1210 X=INT(RND(1)*30)+T4
1220 Y=INT(RND(1)*16)+T4
1225 IF PEEK(D+TD*Y+X)=SP THEN RETURN
1230 GOTO 1210
1500 REM ***  NADALE KUZUSHI ***
1505 Y0=Y:N=TN
1507 POKE D+TD*Y0+X0,SP:Z=Z-T1
1510 FORI=T0TO8:X(I)=TN:NEXT:I=T0:N1=TN
1515 MUSIC"E1GE"
1520 FOR K=TNTOT1:FOR L=TNTO T1
1530 IF PH=PEEK(D+TD*(Y0+K)+X0+L) THENX1=X0+L:Y1=Y0+K:I=I+T1:N1=T1
1535 IF D1=PEEK(D+TD*(Y0+K)+X0+L)THEN N=T1
1540 POKE D+TD*(Y0+K)+X0+L,SP
1560 NEXT L,K
1565 Z=Z-I
1567 IF N1<T0 THEN RETURN
1570 X0=X1:Y0=Y1:GOTO1510
1990 REM
1993 REM *** END ***
1996 REM
2000 PRINT"";
2010 PRINT"♥♥ GAME OVER  ♥♥"
2020 PRINT
2030 PRINT"YOURE SCORE=";INT((Z0-Z)/Z0*100);"%"
2040 PRINT
2060 PRINT
2062 PRINT"RETRY ?? (Y/N)";
2064 INPUT A$
2070 IF A$="Y" THEN 20
2080 END
```

37 シャープ実用プロ2題

MZ-80K

●算数の勉強●

このプログラムはタイトルのように算数の学習プログラムです。それぞれ，九九，1ケタのかけ算，2ケタのかけ算を学習できます。

本プログラムの特徴は画面上のキャラクタの表示位置の制御をカーソルのストリングで処理しているところでしょう。

この方法により，画面に大きな文字（本プログラムにおいては数字）を描かすこともできます。

また，円を描かせたりすることもできます。

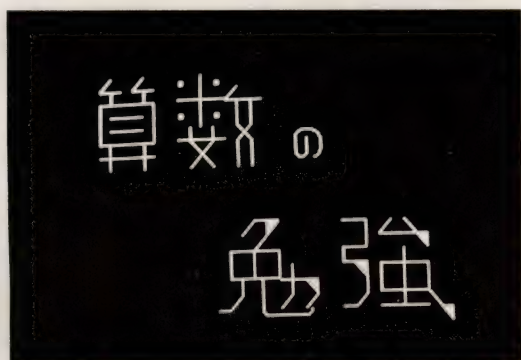
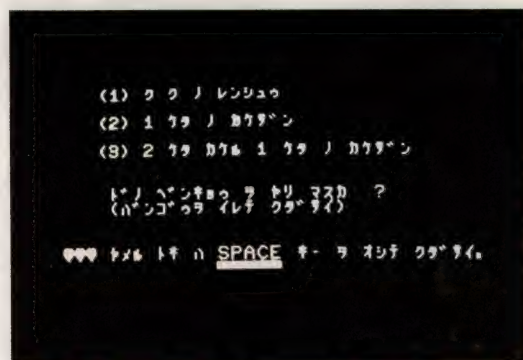
タイトルの「算数の勉強」の文字の表示はPRINT文で制御しています。プログラム（行番号，ステートメント）を組む前に，モニタ画面に表示したい文字，または絵を描いておきます。もちろん行あわせなどはカーソル機能で行ないます。画面に文字または絵を描いてから，行番号とステートメントを編集機能を用いて挿入します。

この機能をうまく使えば，プログラム作成の時間が短縮できます。

プログラムをRUNすると，(1)ククノレンシュウ(2)1ケタノカケザン，(3)2ケタカケル1ケタノカケザンの三種類の入力待ちの状態になります。

終了する場合はスペースキーを押します。それぞれの番号を入力すると算数の勉強のスタートです。プログラムの進行には，GETキーを効果的に使っていますので流れはスムーズです。

計算が正解ならば，●で丸を表示します。もし誤答なら，入力した解答を線で消され，もう一度解答の入力持ちになります。

2ケタかける1ケタについてケタ上りなどがある場合の工夫もおもしろいものです。プログラムに正答数や得点が表示されるようにすると楽しいでしょう。

《写真37－1》初期画面表示

《写真37－2》入力設定

2×3＝□

《写真37—3》
九九の練習問題

○
2×3＝6

《写真37—4》
解答の入力

77
× 8
666

《写真37—5》
まちがった入力

77
× 8
□5 6

《写真37—6》
筆算の型で入力

17
× 8
1 5 6

《写真37—7》
下一ケタをまず計算

17
× 8
136 ○

《写真37—8》
正答

算数の勉強プログラム・リスト

```
10 DIM A$(9)
20 PRINT"■"
30 PRINT"
40 PRINT"
45 PRINT"
50 PRINT"
60 PRINT"
70 PRINT"
80 PRINT"
85 PRINT"■"
90 PRINT"
100 PRINT"
110 PRINT"
120 PRINT"
130 PRINT"
150 PRINT"
160 PRINT"
170 PRINT"
180 A$(0)="
190 A$(1)="
200 A$(2)="
210 A$(3)="
220 A$(4)="
230 A$(5)="
240 A$(6)="
250 A$(7)="
260 A$(8)="
270 A$(9)="
280 B1$="
290 B2$="
300 B3$="
310 B4$="
320 B5$="
330 B6$="
340 B8$="
```

```
345 B9$="┌ ─ ─"
350 S1$="                                  "
355 S2$="────────────────────"
360 S3$="┌────────────────────"
365 S4$="** ﾓｳ ｲﾁ ﾄﾞ ﾖｸ ｶﾝ ｶﾞｴﾃ ﾐﾖｳ｡"
370 D=100:GOSUB 5000
380 PRINT"(1) ｸ ｸ ﾉ ﾚﾝｼｭｳ"
390 PRINT"(2) 1 ｹﾀ ﾉ ｶｹｻﾞﾝ"
400 PRINT"(3) 2 ｹﾀ ｶｹﾙ 1 ｹﾀ ﾉ ｶｹｻﾞﾝ"
410 PRINT"ﾄﾞﾉ ﾍﾞﾝｷｮｳ ｦ ﾔﾘ ﾏｽｶ ?"
420 PRINT"(ﾊﾞﾝｺﾞｳｦ ｲﾚﾃ ｸﾀﾞｻｲ)"
425 PRINT"♥♥♥ ﾔﾒﾙ ﾄｷ ﾊ SPACE ｷｰ ｦ ｵｼﾃ ｸﾀﾞｻｲ｡"
427 PRINT"         "
430 GET B:IF B=0 THEN 430
450 IF B>3 THEN 380
460 PRINT"";A$(B):D=50:GOSUB5000
470 ON B GOTO 1200,1300,500
500 N1=1:TI$="000000"
505 FOR J=1 TO 50
510 GOSUB5200
520 PRINT"";N1
530 PRINT"";A$(M2);"";A$(M1)
540 PRINT "";B2$
550 PRINT"──────────────────────";"";A$(K);
560 ON R GOTO 900,570
570 ON S GOTO 800,580
580 PRINT"";B9$;" ";B8$
590 PRINT"↑";M1;" X";K;
600 GOSUB5300:PRINT"";I
610 GOSUB5300:PRINT"";S1$;"";A$(I);:I1$=STR$(I)
630 PRINT"";B8$
640 PRINT"↑"
650 PRINT"";M2;" X";K;" ﾄ ｸﾘｱｶﾞﾘ｡  "
660 GOSUB5300:PRINT"";S1$;"";A$(I);
665 I3$=STR$(I)
670 GOSUB5300:PRINT"";S1$;"";A$(I):I2$=STR$(I)
680 PRINT"                    "
690 I4$=I3$+I2$+I1$:I=VAL(I4$)
700 IF Q=I THEN PRINT"";B6$:GOTO750
720 PRINT"";S2$
730 PRINT"** ﾓｳ ｲﾁﾄﾞ ﾖｸ ｶﾝｶﾞｴﾃ ﾐﾖｳ｡":D=80:GOSUB5000:GOTO520
750 N1=N1+1:D=10:GOSUB5000:NEXT
760 T=VAL(MID$(TI$,3,2))
765 PRINT "───────────────┐"
767 PRINT "               │"
769 PRINT "───────────────┘"
775 PRINT"";T;" ﾌﾝ ﾃﾞ ﾃﾞｷﾏｼﾀ｡"
780 D=100:GOSUB5000:GOTO380
790 REM ♥♥♥ R=3 S=1
800 PRINT"";B8$;
810 GOSUB5300:I1$=STR$(I)
820 PRINT"";S1$;"";A$(I);
830 PRINT"";B8$;
840 GOSUB5300:I3$=STR$(I):PRINT"";S1$;"";A$(I);"";B8$;
850 GOSUB5300:I2$=STR$(I):PRINT"";S1$;"";A$(I)
860 PRINT"":GOTO690
890 REM ♥♥♥ R=1 S=2
900 IF S=1 THEN GOTO 1100
910 PRINT"";B9$;" ";B8$
920 PRINT"↑";M1;" X";K;
930 GOSUB5300:PRINT"";I
940 GOSUB5300:PRINT"";S1$;"";A$(I);:I1$=STR$(I)
950 PRINT" ";"           ";
960 PRINT"↑";
970 PRINT"";M2;" X";K;" ﾄ ｸﾘｱｶﾞﾘ｡"
980 GOSUB5300:I2$=STR$(I):PRINT" ";"                    "
990 PRINT" ";S1$;"";A$(I)
1000 I3$=STR$(0):PRINT"";:GOTO690
```

```
1090 REM ♥♥♥ R=1 S=1
1100 PRINT"";B8$;
1110 GOSUB5300:I1$=STR$(I):PRINT"";S1$;"";A$(I);
1120 PRINT"";B8$;
1130 GOSUB5300:I2$=STR$(I):PRINT"";S1$;"";A$(I):I3$=STR$(0)
1140 PRINT"":D=50:GOTO690
1200 TI$="000000"
1210 FOR N=1 TO 9
1220 FOR M=0 TO 9
1230 Q=N*M:IF Q<10 THEN S=1:GOTO1240
1235 S=2
1240 PRINT"";A$(N);" ";B2$;" ";A$(M);" ";B5$;"";B8$;
1250 GOSUB5300:I2$=C$:PRINT "";S1$;""; A$(I)
1260 ON S GOTO 1270,1280
1270 PRINT"";:D=20:GOSUB5000:GOTO5500
1280 PRINT"";B8$;:GOSUB5300:I1$=C$:PRINT"";S1$;"";A$(I);
1290 I3$=I2$+I1$:I=VAL(I3$):PRINT"";:GOTO5500
1300 N1=1:TI$="000000"
1310 FOR N=1 TO 50
1320 GOSUB6000
1330 PRINT"";N1;"";A$(K1);" ";B2$;" ";A$(K2);" ";B5$;"";B8$;
1340 GOSUB5300:I2$=C$:ON S GOTO 1350,1400
1350 PRINT"";S1$;"";A$(I)
1360 GOTO 6100
1400 PRINT"";S1$;"";A$(I);
1410 PRINT"";B8$;
1420 GOSUB5300:I1$=C$:PRINT"";S1$;"";A$(I);
1430 I$=I2$+I1$:I=VAL(I$)
1440 PRINT"";:GOTO 6100
5000 D=D-1:IF D<>0 THEN 5000
5010 RETURN
5100 FOR N=1 TO 10
5110 K=INT(RND(1)*9+1)
5120 L=INT(RND(1)*9+1)
5130 NEXT
5140 RETURN
5200 FOR N=1 TO 19
5210 O=INT(RND(1)*99+1):IF O<10 THEN 5210
5220 K=INT(RND(1)*9+1)
5230 NEXT
5240 Q=O*K:IF Q<100 THEN R=1:GOTO5260
5250 R=2
5260 M2=INT(O/10):M1=O-M2*10
5270 IF(M1*K)<10 THEN S=1:GOTO5290
5280 S=2
5290 RETURN
5300 GET C$:IF C$="" THEN 5300
5310 IF C$=" " THEN 380
5320 C=ASC(C$):IFC<48 THEN 5300
5330 IF C>57 THEN 5300
5340 I=VAL(C$)
5350 RETURN
5410 GOSUB5300:PRINT"A$
5500 IF Q=I THEN PRINT"";B6$:GOTO5530
5510 PRINT"";B2$
5520 PRINT"";S4$:D=80:GOSUB5000:GOTO1230
5530 D=20:GOSUB5000
5540 NEXTM:NEXTN
5550 GOTO760
6000 FOR M=1 TO 19
6010 K1=INT(RND(1)*9+1)
6020 K2=INT(RND(1)*9+1)
6030 NEXT
6040 Q=K1*K2:IF Q<10 THEN S=1:GOTO6060
6050 S=2
6060 RETURN
6100 IF Q=I THEN PRINT"";B6$:GOTO6130
6110 PRINT"";B2$
6120 PRINT"";S4$:D=80:GOSUB5000:GOTO1330
6130 D=20:GOSUB5000
6140 N1=N1+1
6150 NEXT
6160 GOTO 760
```

●度数分布表●

入力したデータの度数分布表，平均値，標準偏差，与えられた信頼率で推定するというプログラムです。

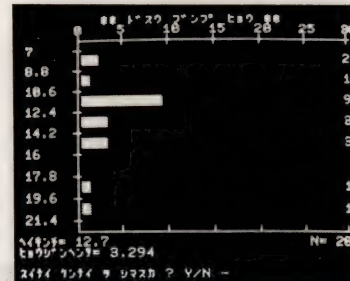
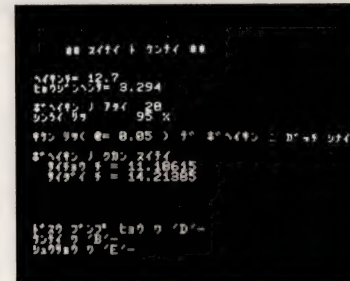
まずデータの量を入力します。これは配列を取るためのものです。データはX－1から順に入力していきます。最初に入力したデータの個数になると画面がクリアされて，度数分布表が表示されます。データの単位はそのつど平均値を中心に処理されますので気にすることはありません。

データに関しての推定をする場合はYを入力してください。まず母平均の値を入力します。この値は母数Nまたは，N－1どちらでもよいでしょう。次に信頼率を入力します。90%か95%を入力します。

そうすると母平均の区間推定がされ，危険率0.05～0.1で最小値，最大値が表示されます。

各ページは，D，Bを入力することにより見ることができます。

《写真37－9》
データの入力

《写真37－10》
度数分布表示

《写真37－11》
検　定

度数分布表プログラム・リスト

```
100 PRINT"▒"
110 PRINT"*** ﾄﾞｽｳ ﾌﾞﾝﾌﾟ ﾋｮｳ ***"
120 PRINT:PRINT
150 INPUT"DATA ﾉ ｶｽﾞ=";N
160 M=0:T=0:Q=0
170 S=99999:L=-99999
200 DIMD(N),F(10),H(10)
210 FORI=1TON
220 N$=STR$(I)
230 PRINT"X-";N$;
240 INPUTX
250 D(I)=X
320 IFX>LTHENL=X
330 IFX<STHENS=X
340 T=T+X
350 Q=Q+X↑2
360 NEXTI
370 R=L-S
380 Y=R/7.5
390 IFR<1THENZ=INT(Y*1000)/1000:GOTO430
400 IFR<10THENZ=INT(Y*100)/100:GOTO430
410 IFR<100THENZ=INT(Y*10)/10:GOTO430
420 Z=INT(Y)
430 S=S-Z
440 FORJ=1TO9
450 S=S+Z
460 H(J)=S
470 NEXTJ
480 FORI=1TON
```

```
490 X=D(I)
500 FORJ=1TO9
510 IFX<H(J)THEN530
520 NEXTJ
530 F(J)=F(J)+1
540 NEXTI
600 PRINT"■":PRINTTAB(10);"** ﾄﾞｽｳ ﾌﾞﾝﾌﾟ ﾋｮｳ **"
610 FORI=0TO30STEP5
620 PRINTTAB(I+6);I;:NEXTI
630 PRINT:PRINTTAB(7);"├";
640 FORJ=1TO6
650 PRINT"────┼";:NEXTJ:PRINT
660 FORK=1TO8
670 PRINTTAB(7);"┼":PRINTTAB(7);"|"
680 NEXTK
690 PRINTTAB(7);"┼"
700 PRINTTAB(7);"└";
710 FORL=1TO6
720 PRINT"────┴";:NEXTL
750 PRINT"▒▒▒▒"
760 FORJ=1TO9:PRINTH(J):PRINT:NEXTJ
770 PRINT"▒▒▒"
780 FORI=1TO9
790 IFF(I)=0THEN840
800 FORJ=1TOF(I)
810 PRINTTAB(8);"▒";:IFJ>27THEN830
820 NEXTJ
830 PRINTTAB(36);F(I);
840 PRINT:PRINT:NEXTI
850 PRINT
860 M=T/N
870 V=(Q-N*M↑2)/(N-1)
880 S=SQR(V)
890 PRINT"ﾍｲｷﾝﾁ=";INT(M*1000)/1000;
900 PRINTTAB(33);"N=";N
910 PRINT"ﾋｮｳｼﾞﾝﾍﾝｻ=";INT(S*1000)/1000:PRINT
920 PRINT"ｽｲﾃｲ ｹﾝﾃｲ ｦ ｼﾏｽｶ ? Y/N - ";
930 GETN$
940 IFN$="Y"THEN970
950 IFN$="N"THEN2000
960 GOTO930
970 PRINT"■":PRINT"    ** ｽｲﾃｲ ﾄ ｹﾝﾃｲ **":PRINT:PRINT
980 PRINT"ﾍｲｷﾝﾁ=";INT(M*1000)/1000
990 PRINT"ﾋｮｳｼﾞﾝﾍﾝｻ=";INT(S*1000)/1000:PRINT
1000 INPUT"ﾎﾞﾍｲｷﾝ ﾉ ｱﾀｲ  ";MM
1010 INPUT"ｼﾝﾗｲ ﾘﾂ      ";P:PRINTTAB(16);"■%"
1020 IFP<50THENPRINTTAB(18);"■?":GOTO1010
1030 MX=ABS(M-MM):T0=MX/SQR(V/N)
1040 A=1-(P/100)
1050 Z=-LN(2*A)*(1-A/2)
1060 ZQ=(Z*(2.0611786-5.7262204/(Z+11.640595)))↑(1/2)
1070 TQ=ZQ*((1-1/(4*(N-1)))↑2-1/(2*(N-1))*ZQ↑2)↑(-1/2)
1080 IFT0>=TQTHEN1100
1090 PRINT:PRINT"ｷｹﾝ ﾘﾂ( @=";A;" ) ﾃﾞ ﾎﾞﾍｲｷﾝ ﾆ ｶﾞｯﾁ ｽﾙ":GOTO1110
1100 PRINT:PRINT"ｷｹﾝ ﾘﾂ( @=";A;" ) ﾃﾞ ﾎﾞﾍｲｷﾝ ﾆ ｶﾞｯﾁ ｼﾅｲ"
1110 PRINT:PRINT"ﾎﾞﾍｲｷﾝ ﾉ ｸｶﾝ ｽｲﾃｲ "
1120 PRINT"  ｻｲｼｮｳ ﾁ =";M-TQ*SQR(V/N)
1130 PRINT"  ｻｲﾀﾞｲ ﾁ =";M+TQ*SQR(V/N)
1200 PRINT:PRINT:PRINT:PRINT:PRINT"ﾄﾞｽｳ ﾌﾞﾝﾌﾟ ﾋｮｳ ﾜ 'D'- "
1210 PRINT"ｹﾝﾃｲ ﾜ 'B'- "
1220 PRINT"ｼｭｳﾘｮｳ ﾜ 'E'- "
1230 GETW$
1240 IFW$="D"THEN600
1250 IFW$="B"THEN970
1260 IFW$="E"THEN2000
1270 GOTO1230
2000 END
```

38 CLOADマガジンより

I.ロードラリー
II.ハッスル

TRS-80LII

庄司　渉

ここに御紹介する二つのTRS80LEVELII用のゲームプログラムは，㈱エー・アンド・エー・ジャパン社より，CLOADマガジンとして，販売または販売予定のソフトウエア・パッケージ（カセット）に納められているものです。このような形で発表することができるのは，同社の御好意によるものに他なりません。なお，この二つのプログラムは，別々のカセットに納められていますが，一つのカセットには，数個のゲーム等のプログラムがLEVELI，LEVELII別に納められています。

●ロード・ラリー●

第37－1表のプログラムリストはスピードウェイ）カセットの表示では，ロード・ラリーとなっている）という名がついています。

さて，リストをみながらプログラムを入れる段階で，多少の英文が読める人なら，このゲームがどんなもので，どうやって遊ぶのかはおわかりでしょう。

英語があまり好きでない人のために一言だけ説明しておきましょう。

このプログラムをRUNさせると，このゲームの内容や遊び方が表示されます。わけがわからなければ〈ENTER〉を2回押します。すると画面にサーキットと，小さな車が現われてきます。

〈ENTER〉キーを押すとスタートです。スピードは2段切りかえができ，それはスペースキーでおこないます。

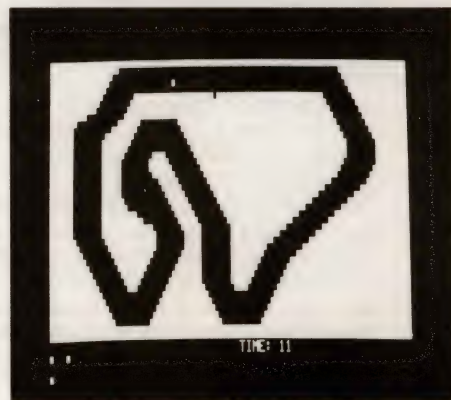
道路を左斜め前方にかえる時には，Qキーを押します。右斜め前方にかえたい時には⓪キーを押します。

カベに激突したときには，あわてず様子を見た方がよいでしょう。

一周に要した時間が表示され，それまでの最高タイムと比較されます。

《写真38―1》◀

《写真38―2》◀

《第37－1表》ロードラリー

```
10  CLS.PRINT.PRINTTAB(17),"* *  S P E E D W A Y  * *":PRINT.PRINT
15  PRINT"     THIS PROGRAM WAS MADE BY PAUL PETERSON OF"
20  PRINT"WINTHROP, MINNESOTA."
22  PRINT"     IN THIS GAME OF SPEEDWAY, A TWISTING TRACK IS"
24  PRINT"PRINTED OUT ON THE SCREEN AND YOUR CAR WILL APPEAR"
26  PRINT"AT THE TOP OF IT AS A SMALL LIGHT."
28  PRINT"     THE OBJECT OF THE GAME IS TO GET YOUR CAR"
30  PRINT"AROUND THE TRACK IN THE LEAST TIME.  ONCE THE"
35  PRINT"CAR IS MOVING, YOU CAN SHIFT BETWEEN A FAST AND"
40  PRINT"SLOW SPEED BY PRESSING THE  SPACE  BAR."
42  PRINT
46  INPUT" PRESS  ENTER  TO CONTINUE",A$.CLS.PRINT.PRINT
48  PRINT"     YOU DRIVE THE CAR BY PRESSINNG THE  Q  KEY"
50  PRINT"FOR LEFT TURNS AND THE  @  KEY FOR RIGHT TURNS."
55  PRINT"PLACE YOUR FINGERS ON THESE KEYS NOW SO YOU GET"
60  PRINT"USED TO THEIR POSITIONS.  JUST SO YOU DON'T GET"
62  PRINT"CONFUSED WHEN THE CAR IS MOVING DOWN THE SCREEN."
64  PRINT"TRY TO IMAGINE YOU'RE IN THE CAR MAKING THE TURNS."
66  PRINT"OTHERWISE YOU MAY PRESS THE WRONG KEYS."
68  PRINT"     IF YOU DO, AND YOU HIT THE CURB, YOUR CAR"
70  PRINT"WILL BOUNCE OFF AND AUTOMATICALLY SHIFT INTO SLOW"
75  PRINT"SO YOU CAN GET GOING IN THE RIGHT DIRECTION AGAIN."
80  PRINT.PRINTTAB(17);.INPUT"PRESS  ENTER  TO BEGIN",A$
100  X=50.Y=3.R=5.H=150.L=10000
110  CLS.PRINT@897,"",.SET(0,44):SET(5,44):SET(0,47):FORB=0TO41
120  READ P,Q.FORA=PTOABS(Q).SET(A,B):NEXTA.IFQ<109GOTO120
130  NEXTB.RESET(51,5).SET(X,Y):T=0.GOTO 3010
135  PRINT@925,"TIME:",INT(T),.PRINT@897,"",.IF(X=51) AND(Y<6)AND(T>70)GOTO3000
140  IF(POINT(0,44)=0)AND(POINT(5,44)=0)AND(POINT(0,47)=0)GOTO110
141 I$=INKEY$.IFI$<>" "GOTO150
146  IFH=0 H=150.GOTO 149
147  H=0
149  PRINT@897,"",
150 IFI$="@"THEN R=R-1.PRINT@897,"",
160 IFI$="Q"R=R+1.PRINT@897,"",
170  IFR>8R=R-8
180  IFR<1R=R+8
190  ON R GOTO 200,210,220,230,240,250,260,270
200  A=1.B=0.GOTO280
210  A=1:B=-1.GOTO280
220  A=0.B=-1:GOTO280
230  A=-1:B=-1.GOTO280
240  A=-1:B=0.GOTO280
250  A=-1:B=1.GOTO280
260  A=0.B=1.GOTO280
270  A=1.B=1
280  IFPOINT(X+A,Y+B)=-1GOTO300
290  RESET(X,Y).X=X+A:Y=Y+B.SET(X,Y)
293  FORW=1TOH.NEXTW.T=T+1.IFH=0T=T-.5
296  GOTO135
300  PRINT@970,"WATCH OUT!!  YOU HIT THE CURB!",
310  PRINT@ 897,"";.FORW=1TO750:NEXTW.PRINT@970,"
320  PRINT@897,"                    ";.R=R-5.H=150:GOTO135
3000  RESET(X,Y).PRINT@216,"LAST LAP:",T;.X=50.H=150
3005  IFT<LTHEN L=T:PRINT@280,"BEST LAP:",L;
3010  SET(X,Y):T=0.PRINT@905,,,,"                ".PRINT@905,"PRESS  ENTER  TO START",
3020  INPUTA$.PRINT@905,,,,"               ".PRINT@897,"",.SET(0,47).GOTO135
7000  DATA 0,110,0,20,90,110,0,18,91,110,0,17,92,110,0,16,93
7010  DATA 110,0,15,26,84,94,110,0,14,25,85,95,110,0,13,24,86
7020  DATA 96,110,0,10,23,87,97,110,0,7,21,27,40,88,98,110
7030  DATA 0,6,19,26,41,89,99,110,0,6,17,25,42,90,100,110
7040  DATA 0,6,16,24,43,91,101,110,0,6,16,23,44,91,101,110,0
7050  DATA 6,16,22,32,35,45,91,101,110,0,6,16,21,31,36,46,90
```

```
7060  DATA 101,110,0,6,16,21,31,37,47,89,100,110,0,6,16,21,32
7070  DATA 38,48,88,99,110,0,6,16,21,34,39,49,86,98,110,0,6,16
7080  DATA 21,35,40,50,84,96,110,0,6,16,21,36,41,51,82,94,110
7090  DATA 0,6,16,22,37,42,52,80,92,110,0,6,16,23,38,43,53,78
7100  DATA 90,110,0,6,16,26,39,44,54,76,88,110,0,6,16,28,40,45
7110  DATA 55,74,86,110,0,6,16,31,41,46,56,72,84,110,0,6
7120  DATA 16,32,42,46,56,70,82,110,0,7,17,32,42,46,56,68
7130  DATA 80,110,0,8,18,32,42,46,56,66,78,110,0,9,19,31,41
7140  DATA 46,56,65,76,110,0,10,20,30,40,46,56,64,74,110
7150  DATA 0,11,21,29,39,46,56,63,73,110,0,12,22,28,38,46
7160  DATA 56,62,72,110,0,13,23,27,37,46,56,61,71,110,0,14
7170  DATA 24,26,36,47,57,60,70,110,0,15,35,48,69,110,0
7180  DATA 16,34,49,68,110,0,17,33,50,67,110,0,18,32,51,66
7190  DATA 110,0,19,31,52,65,110,0,110,0,110
7200  END
```

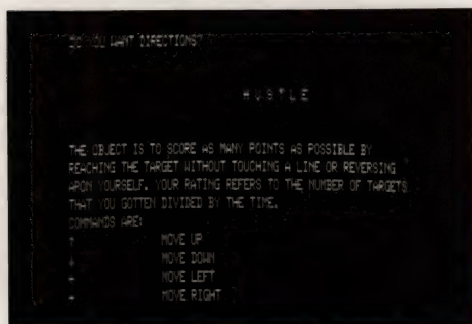
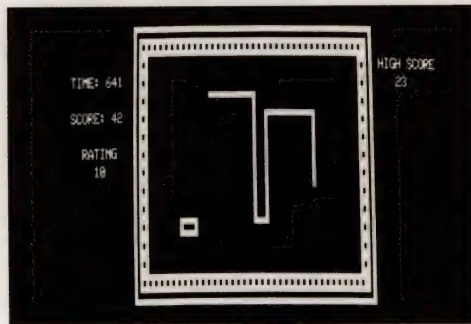
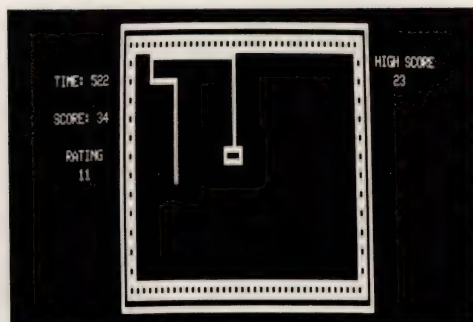
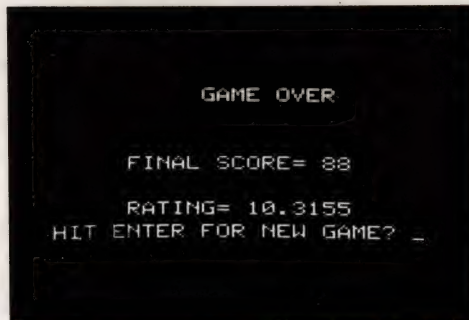
●ハッスル●

第37－2表のプログラムは，ハッスルと呼ばれるゲームです。

四角形の大きなわくの中に，ヘビのようなものと，そのエサに当るようなものが出てきます。このヘビをエサに向ってコントロールします。コントロールする方向と⇧キー，⇩キー，⇦キー，⇨キーが同じむきで対応しています。

まわりのワクにぶつけたり，進行方向と逆の方向のキーを押したり，あるいは，自分自身の体にぶつけた時点で，ゲームは終りとなります。

エサには点がついていて（それは，食いついてみるまではわからないのですが）ヘビをそれにぶつけると，点が加算されていきます。と同時に，その点の大きさに比例して，オッポが長くなっていくのです。体が長くなればなるほど，そのコントロールはむつかしくなります。

なかなかおもしろいゲームで，時間がたつのを忘れます。徹夜してしまいましたが，最高193点（RATING＝123,379）しかとれませんでした。ぜひ挑戦してみてください。

《写真38－3》

《写真38－4》

《写真38－5》

《写真38－6》

《第38－２表》ハッスル・ゲーム・プログラム・リスト

```
1 '    -- H U S T L E --
2 'WRITTEN BY PETER TREFONAS, U.N.O. (12/6/78)
3 GOTO20
10 RANDOM.CLS.DEFINT I,F,X,Y.A$="":IZ=0.IP=1:IN=-1:X=62:Y=24:X1=1:Y1=0:F3=0:F2=0:F0=-3.F4=0:IS=0.Q=0:I0=0:IB=-1.P=0:P1=.9:IT=550.IQ=
0.F7=12.F8=79.FZ=32.FY=15360.FX=15359.FW=15361:I8=0.FV=128.F9=64.RETURN
20 GOSUB10.DIM I(IT,1):INPUT"DO YOU WANT DIRECTIONS";A$.IFA$="NO"ORA$="N"GOTO190
23 PRINT.PRINT.PRINTTAB(30),"H U S T L E".PRINT:PRINT
26 PRINT"THE OBJECT IS TO SCORE AS MANY POINTS AS POSSIBLE BY
29 PRINT"REACHING THE TARGET WITHOUT TOUCHING A LINE OR REVERSING
30 PRINT"APON YOURSELF. YOUR RATING REFERS TO THE NUMBER OF TARGETS
31 PRINT"THAT YOU GOTTEN DIVIDED BY THE TIME.
32 PRINT"COMMANDS ARE.
35 PRINTCHR$(91),"MOVE UP
38 PRINTCHR$(92),"MOVE DOWN
41 PRINTCHR$(93),"MOVE LEFT
44 PRINTCHR$(94),"MOVE RIGHT
48 FORI=1TO4000.NEXT
190 GOSUB8000.'DRAW SCREEN
200 GOSUB1000.GOSUB2000.GOSUB3000.GOTO200
800 FORI0=1TO4.NEXT.RETURN
999 'NEXT SBR. PROCESSES COMMANDS
1000 A$=INKEY$.IFA$=""RETURN
1010 IFASC(A$)=8LETX1=IN.Y1=IZ.RETURN
1020 IFASC(A$)=9LETX1=IP.Y1=IZ.RETURN
1030 IFA$="["LETX1=IZ.Y1=IN.RETURN
1040 IFASC(A$)=10LETX1=IZ.Y1=IP.RETURN
1060 RETURN
1999 'NEXT SBR. RESETS POINTERS AND MOVES DOT
2000 X=X+X1.Y=Y+Y1.F3=F3+IP.IFF3>ITLETF3=IZ.IQ=IQ+IT
2020 IFF0<IZLETF0=F0+IP.GOTO2040
2030 F2=F2+IP.IFF2>ITLETF2=IZ
2040 RESET(I(F2,IZ),I(F2,IP)).I(F3,IZ)=X.IFPOINT(X,Y)THEN2500
2050 SET(X,Y).I(F3,IP)=Y.RETURN
2500 P=INT(Y/3)*64+INT(X/2).IFP-1=F4ORP=F4ORP+1=F4THEN2520.ELSE2600
2520 Q=RND(9).IS=IS+Q.F0=F0-Q*2.I8=I8+1
2530 FORI=IPTO8.OUT255,4.PRINT@F4+IN,STR$(Q);" ";.GOSUB800.OUT255,3.PRINT@F4+IN,CHR$(183),CHR$(179),CHR$(187);:NEXTI
2540 PRINT@F4-IP,"   ";.F4=IP.IB=IN.PRINT@197,IQ+F3;.PRINT@326,IS;.PRINT@515,INT(I8*1000/(IQ+F3));:GOTO2050
2599 'YOU LOSE HERE
2600 CLS.PRINTCHR$(23);.PRINT.PRINT:PRINT
2610 PRINTTAB(10),"GAME OVER".PRINT.PRINT.PRINTTAB(5),"FINAL SCORE=";IS.PRINT.PRINTTAB(5),"RATING=",I8*1000/(IQ+F3):FORI=1TO1000.NEX
T
2620 FORI=IZTOTI.I(I,IZ)=IZ.I(I,IP)=IZ.NEXT
2630 IFIS>HILETHI=IS
2635 INPUT"HIT ENTER FOR NEW GAME",A$
2640 GOSUB10.GOTO190
2999 'THIS SUBR PUTS BOXES ON SCREEN
3000 IFIB>IZLETIB=IB-IP.RETURN
3010 IFIB=IZPRINT@F4-IP,"   ";.IB=IN
3020 IFRND(IZ)<P1RETURN
3025 F4=RND(F7)*F9+F8+RND(FZ)
3026 IFPEEK(FX+F4)<>FZANDPEEK(FX+F4)<>FVTHEN3025
3027 IFPEEK(FY+F4)<>FZANDPEEK(FY+F4)<>FVTHEN3025
3028 IFPEEK(FW+F4)<>FZANDPEEK(FW+F4)<>FVTHEN3025
3030 PRINT@F4-1,CHR$(183);.PRINT@F4,CHR$(179);.PRINT@F4+1,CHR$(187);.IB=RND(65)+25.RETURN
7999 'FOLLOWING SUBR DRAWS SCREEN
8000 CLS.FORI=1TO46.SET(22,I).SET(103,I).NEXT
8010 FORI=3TO44.SET(26,I).SET(99,I).NEXT
8020 FORI=23TO102.SET(I,IP).SET(I,46).NEXT
8030 A$=CHR$(183).FORI=76TO114.PRINT@I,A$;.NEXT
8040 FORI=908TO946.PRINT@I,A$;.NEXT
8050 FORI=2TO13.PRINT@64*I+12,A$;.PRINT@64*I+50,CHR$(187);.NEXT
8060 SET(101,4).SET(101,43)
8070 PRINT@192,"TIME.";.PRINT@320,"SCORE.";:PRINT@448,"  RATING";.PRINT@181,"HIGH SCORE";:PRINT@247,HI;:RETURN
```

第4章

マイコン応用資料

大型プログラム実行の為に

マイコンのメモリー容量拡張

NEC　　TK-M20Kの接続

米　持　尚

マイクロコンピュータのプログラムも，しだいに大型のものになってくると，まず第一につきあたる問題はメモリーの容量の点です。ＤＩＭ文などを使った大量データ処理をやろうとすると，５～７Ｋバイト程度ではすぐメモリオーバーフローを起こしてしまい，せっかくのマイコンの機能が十分生かせなくなってしまいます。

その点，この外部ＲＡＭ／ＲＯＭボードは，各種のＩ／Ｏインターフェースをそなえた手ごろな拡張用ボードです。すでにＴＫ－80，ＴＫ80ＢＳを使用していて，本格的にマイコンを使いこなそうという意欲をお持ちの方はシステム拡張の場合はまずこのボードを検討するとよいでしょう。

ＴＫ－Ｍ20Ｋ

ＴＫ－Ｍ20ＫはＴＫ80，ＴＫ80ＢＳと同寸法のボード１枚に

○ＲＡＭ12ＫＢ（μＤＰ2114×24個）
○ＲＯＭ用ソケット（μＰＤ458×８個：８Ｋバイト分のソケットを実装）
○Ｉ／Ｏ（μＰＤ8255×２個，μＰＤ8251×１個）
○バスバッファ（μＰＢ8216×４個）
○制御回路

という構成でまとめあげられています。ＴＫ80＋ＴＫ80ＢＳ＋ＴＫＭ20Ｋという組合せにすることにより，ＲＡＭは20Ｋバイト，ＲＯＭは９Ｋバイトまで拡張されます。ただし購入時，ＲＯＭは基板上ではソケットのみ)。

このＴＫ－Ｍ20Ｋの特徴としては，メモリマッピングの融通性，ＲＡＭのメモリ内容をプログラム自体によって破壊してしまわないＲＡＭプロテクト機能を有する点などがあげられます。

メモリマッピングの融通性については，ＲＯＭ，ＲＡＭ，Ｉ／Ｏはアドレッシング上五つのブロックに分割されています。ＲＡＭ12Ｋバイトを４Ｋバイトごとに３分割してＸ，Ｙ，ＺブロックとしＲＯＭをＶブロック，Ｉ／ＯをＷブロックとし，それぞれのブロックにはアドレスセレクタが付いています。

このアドレスセレクタ（ジャンパー線で行なう）の設定をいろいろと変えることによって各ブ

ＴＭ－Ｍ20Ｋの仕様

ＲＡＭ容量	：12288バイト （μＰＤ2114×24)実装
ＲＯＭ容量	：8192バイト （μＰＤ458×８用ソケットのみ実装）
パラレル Ｉ／Ｏポート	：48ポート（８×３×２）ポート （μＰＤ8255×２）実装
シリアル Ｉ／Ｏポート	：μＰＤ8251×１　実装
電源電圧	：外部電源必要 ＋５Ｖ±５％　＋12Ｖ±５％
消費電流	：＋５Ｖ→3.1Ａ(ＭＡＸ) ＋12Ｖ→0.5Ａ(ＭＡＸ) （μＰＤ458も実装したフルメモリでの値です。）
基板寸法	：310×180㎜ （ＴＫ－80Ｅ／80と同じサイズです。）
付　属　品	：取り付け部品一式（マザーボード，100ピンコネクタは含みません。）

ロックをメモリ領域の中の任意のアドレスに配置するとこができるのです。ですからメモリマップの異なる応用例に対しても，自由に対応することが可能です。

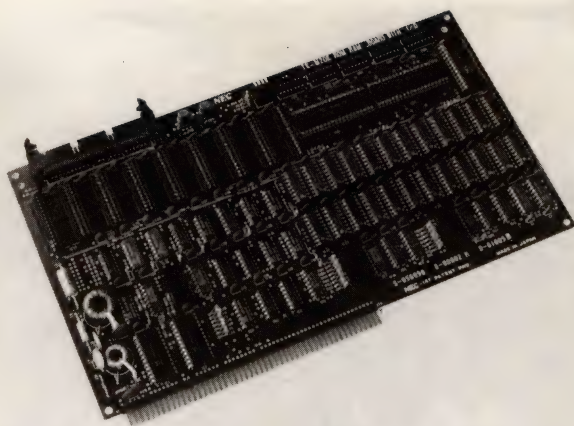
又，RAMプロテクト機能は，プログラム開発時にいったんプログラムを書込んだあとRAMを読み出し専用に切替えておけば，ちょっとした動作ミスや，プログラムの暴走などによりせっかくのプログラムを消してしまうということもなくなります。特にPROMを使ってプログラム開発するより機能をかなりアップできそうです。

《写真1》TK－M20Kボード上のようす

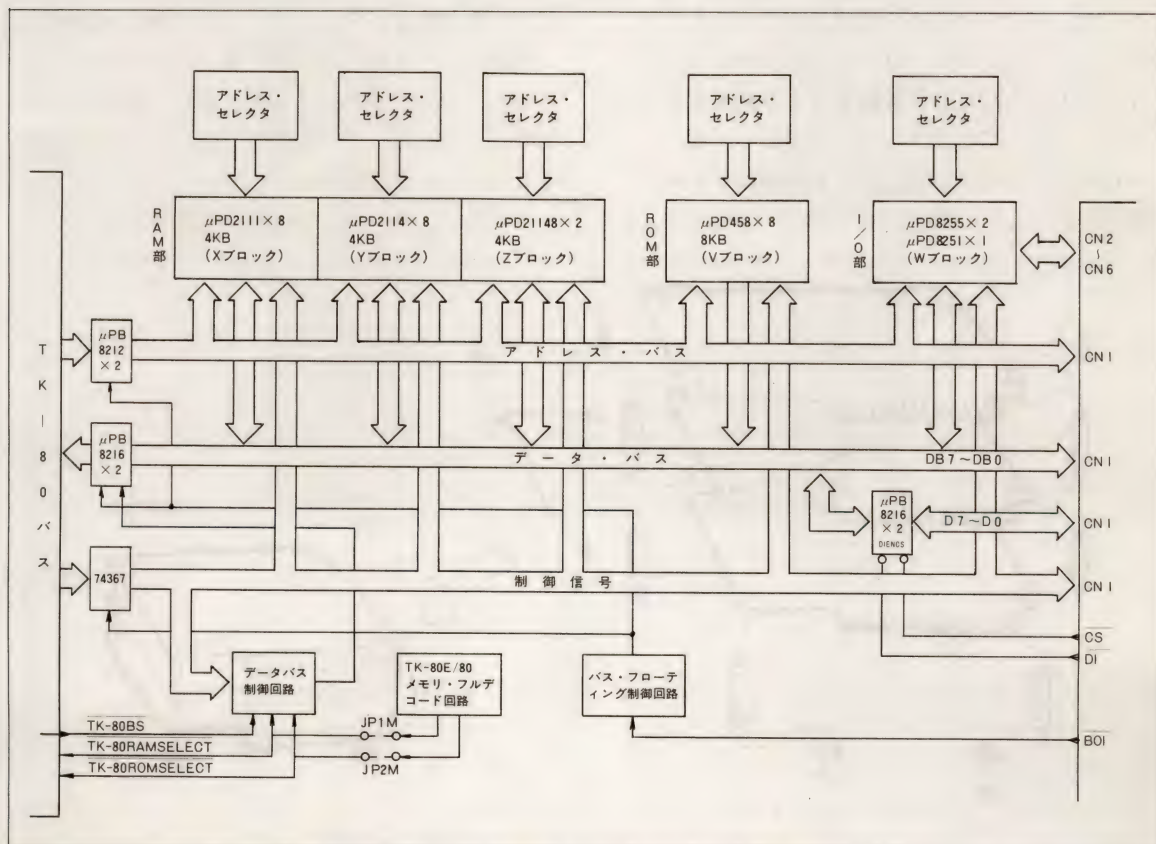
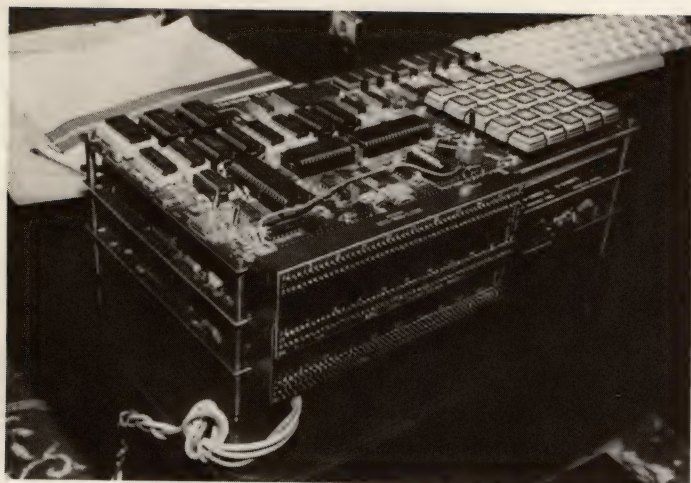
TK80BSシステムとの組合せ

本誌に登場するいくつかのプログラムはTK－M20Kを使用しています。参考までにTK80，BSとTK－M20Kの接続の際の手順を紹介しておきましょう。

TK－80，BS上の変更

TK－80とTK－80BSを組合わせて使っている方は，その時点でTK－80上にいくつかの加工を行なっています。それにさらに3カ所のジャンパー線を追加します。

それは，

①TK－80μPB8228の25番ピンと基板の部品面の20番ピン（A－20）

②TK－80のμPB8228の27番ピンと基板の部品面21番ピン（A－21）

《第1図》TK－M20Kのブロック・ダイヤグラム

《写真2》TK80，TK－80BSとの組合せ

③TK－80のμPB8224の6番ピンと基板の部品面49番ピン（A－49）
の3カ所です。

又，TK80－BSの基板上では1個所のジャンパー線が必要です。それは，

TK－80BSのμPD2332-037の右隣りのμPB8216の1番ピンと基板の半田面の47番ピン（B－47）です。

ここで，本誌のはじめに掲載されているLEVEL－I，LEVEL－IIの改造をすでになされた方は，TK－80及びTK80BSのB47番ピンを使用してしまっていますので，このLEVEL－I↔LEVEL－II切替えのための線はTK80，TK80BSとも，となりのB－46へうつします。

こうすればTK－M20Kを追加した後でもLEVEL I↔LEVEL IIの切替えが従来どおり行なえます。

アドレスセレクタ

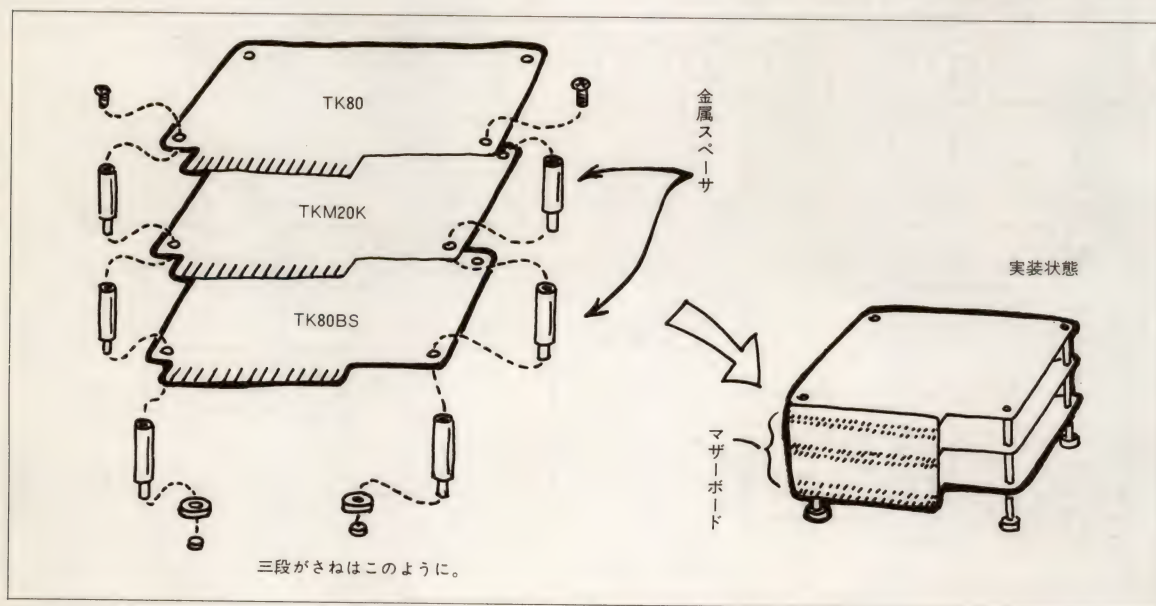
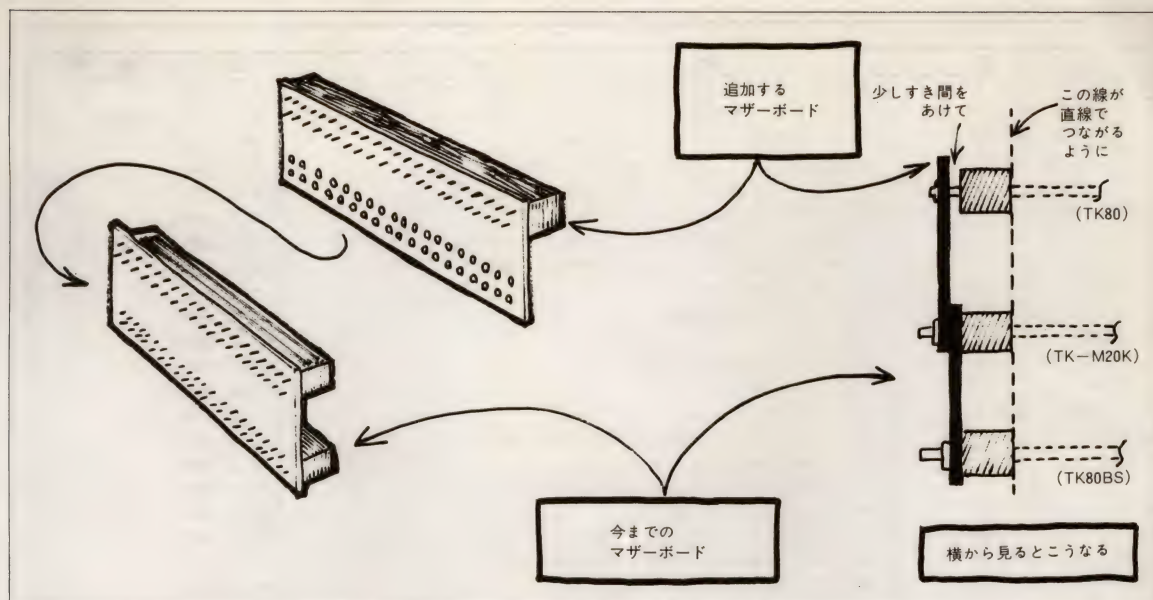
TK－M20Kの基板上にはRAM，I／Oなどのアドレスセレクタ（ジャンパーソケット），RAMプロテクトスイッチなどがあり，使用に際して，アドレスセレクタの部分は何カ所かハンダ付けする必要があります。標準仕様の場合は，8カ所，説明書にも配線部分が示されています。また，各種のアドレス設定も一覧表が説明書に掲げられています。

RAMプロテクトのスイッチは一応全部下側へたおしてききます（OFF）。

マザーボードの配線

TK－80，TK－80BS，TK－M20Kの三つの基板をかさねあわせるため各部の接続に三段のマザーボードが必要です。

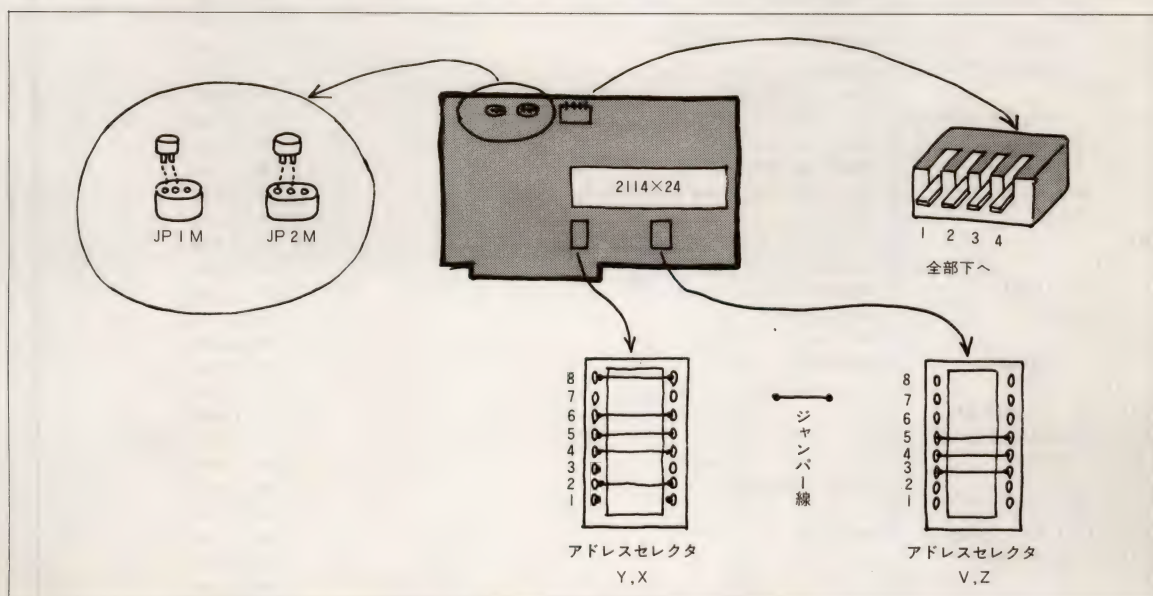
その接続用ボード，ソケットはTK－M20Kには付属していませんので別に購入しなければなりません。マザーボードはTK－80↔TK80BS接続用のものと同一で，従来の二段ソケットに追加する形でハンダ付けします。

この場合，あらかじめ3枚の基板を付属スペー

《第2図》三段がさねにして使用する

《第3図》マザーボードの加工はこのように

サーで三段に組立てておき，そこにマザーボードを差込んで，実装状態にしてハンダ付けしたほうが歪みが出ません。別々に組んで，後から差込むより，後々スムースに脱着ができるようになります。

幅広い応用を

以上でメモリー拡張は終了です。通常の使用状態ではＲＡＭの追加分のアドレス領域はＸブロックがA000〜ＡＦＦＦ，ＹブロックがB000〜ＢＦＦＦ，ＺブロックがC000〜ＣＦＦＦとなっています。ＢＡＳＩＣスタート時のＲＡＭＥＮＤ？に対してはＥＦＦＦと入力するわけです。これで，かなり大きなプログラムもＳＩＺＥを気にせず伸び伸び組んだり入力したりすることができるようになりました。

しかし，このＴＫ－Ｍ20Ｋは単なる拡張ＲＡＭボードではありませんから，その豊富なＩ／Ｏボード，ROMを大いに活用して手持ちのマイコンを本格的なシステムにしあげるために活用してください。

《第4図》ＴＫ－M20Ｋのセット状態

■市販主要ソフトテープ

ソフト名	内容	価格	備考
スロットマシン	BASICとマシン語をリンクしハイスピードでドラム回転	¥ 2,500	
ボーリング	あなたに代ってマイコンがシュート	¥ 2,500	MZ80K
ローン計算	借金、不動産、ローンなどの計算	¥ 2,800	
ヤシの実落し	決死のジャンプ　ワニに食べられないように	¥ 2,500	
オセロ	マイコンとオセロ・ゲームの対決	¥ 2,500	
ブロックくズシ	おなじみのゲームセンターが手元で！	¥ 2,500	
マージャン	役満をねらえ！振り込みにご注意！	¥ 3,000	BASIC
水泳	何コースを泳ぎますか　5人で順位を競おう	¥ 2,500	
バリケード	障害物に気をつけて	¥ 2,500	
スタートレック	おなじみの宇宙大戦争	¥ 2,800	
価値判定	あなたに代ってすべてを判定	¥ 3,000	(ハドソン)
ベースボール	あなたの力で球団を経営	¥ 2,800	
パチンコ	おなじみパチンコ	¥ 3,000	
サルも木から落ちる	サルの木登りじゃまをして下さい	¥ 2,600	
陣取りゲーム	コンピューターと画面上の陣地を競います	¥ 2,600	
チェッカー	チェス板でおこなうゲーム	¥ 2,800	
ポーカー	カードゲームの王様・ポーカーをマイコンで	¥ 3,000	
雀球	マージャン・パチンコ	¥ 3,000	
野球拳	おなじみのヨヨイノヨイ	¥ 2,800	
ブラックジャック	マイコンが親になってゲームを進めます	¥ 3,000	
ダービー	競馬ゲーム	¥ 2,800	
英会話レッスン	英会話によく使う基本表現を学習	¥ 2,800	
スーパーゴルフ	ゴルフのプロコース全18ホールをプレイ	¥ 3,800	
ハングマン	首つりゲーム	¥ 2,800	
D-DAY	史上最大の作戦	¥ 3,000	
GALインタプリタ	GAMEをMZ80K用に拡張	¥ 5,000	MZ80K
データベース	電話帳、住所録等簡単な在庫管理に使用	¥ 2,800	
リナンバー	BASICの文番号を10番おきにそろえる	¥ 3,000	マシン語
プリンター用画面コピー	画面全体をそっくりコピーすることが可能	¥ 2,500	
RAM TEST	何列目の何番目のRAMが不良かを知らせてくれる	¥ 2,500	(ハドソン)
スペースアクロバット	宇宙空間を無重力でジャンプ	¥ 2,800	MZ80K
ブロックくズシ	インベーダーブロック	¥ 2,500	
シーウルフ	潜水艦から敵の船を攻撃する	¥ 3,000	
ピンボール	左右2個のフリッパーでボールをコントロール	¥ 2,800	
ヘッドオン	コンピューターの追跡をかわしながらコースを通過	¥ 2,800	
オリンピック	陸上競技	¥ 3,000	GAL
ガンマン	マイコンと広野の決闘！	¥ 2,800	
マシン語CAI	Z-80マシンランゲージを図解で説明	¥ 2,500	(ハドソン)
カエルのジャンプ	カエルがなきながらハエを取るゲーム	¥ 2,500	
ボクシング	コンピューターとボクシング	¥ 3,000	
LUNAR LANDER	月面軟着陸ゲーム	¥ 2,500	
BLACKJACK	トランプゲーム	¥ 3,000	
SPACE WARS	PET版スタートレック	¥ 3,500	
SPACE TALK/SPACE FIGHT	宇宙戦争2人用	¥ 3,500	
MORTGAGE	ローン返済計算	¥ 4,000	
MACHINE LAMGUAGE MONITOR	マシーン語プログラム	¥ 3,000	
DISASSEMBLER	逆アッセンブラ	¥ 1,000	
BASIC BASIC	PET・BASICの学習	¥ 3,000	
BASE BALL	巨人ー阪神戦、投打共コントロール可能	¥ 3,000	
TREK-2001	スタートレックの機械語版	¥ 3,000	
GRP	グラフによる多次方程式の解を求める	¥ 3,000	
MATRIX	行列式の演算	¥ 3,000	
SEESAW JUMP	風船割りゲーム	¥ 3,000	
SUBMARINE	潜水艦爆破ゲーム	¥ 2,000	
INVADER	PET版インベーダー	¥ 3,000	
SUPER GOMOKU	機械語版ゴモクナラベ	¥ 3,000	
LAND SLIDE	落下物捕獲ゲーム	¥ 3,000	
LINKAGE PROGRAM	リンケージプログラム	¥ 1,000	
PET ASSEMBLER	ペットアセンブラー	¥ 3,000	PET
MORSE CODER	モールスコードの練習	¥ 3,000	
ANALYSIS	データの統計処理	¥ 3,000	
BRICK BREAK	ブロックくずし	¥ 3,000	
PET CONCENTRATION	神経衰弱	¥ 2,500	
SUPER BOWLING	スーパーボーリング	¥ 2,500	
PET SLOT	スロットマシン	¥ 3,000	
MASTER MIND	色あてゲーム	¥ 3,000	

ソフト名	内容	価格	備考
SQUIGGLE	ランダム関数プログラム演習用	¥ 2,500	
TRIG	ピタゴラス定理教育用	¥ 2,000	
TIC-TAC-TOE	三目並べゲーム	¥ 2,000	
ROTATE	文字並べゲーム	¥ 2,000	
OTHELLO	オセロゲーム	¥ 2,000	
TARGET PONG	ボールゲーム	¥ 2,000	
OFF-THE-WALL	ボールゲーム	¥ 2,000	
DEATH STAR	撃墜ゲーム	¥ 2,000	
REVERSE	数字並べゲーム	¥ 2,000	
BIORHYTHM	バイオリズム	¥ 2,000	
DRAW POKER	トランプゲーム	¥ 2,000	
UFO SHOOTING	宇宙ゲーム	¥ 2,000	
DIET PLANNER	痩身計画	¥ 2,000	
AMORTIZATION	経理計算演習応用例	¥ 2,000	
GUESSING GAME	数当てゲーム	¥ 2,000	PET
MATH TEACHER	四則演算演習プログラム	¥ 2,000	
CAR RACE	自動車レースゲーム	¥ 2,000	
BOWLING	ボーリングゲーム	¥ 1,500	
BARRICADE GAME	ヘビによるカエル喰いゲーム	¥ 1,500	
CONCENTRATION	モグラたたきゲーム	¥ 2,000	
GOLF	ゴルフゲーム	¥ 2,000	
SUPER ROULETTE	本格派ルーレット	¥ 2,000	
PICK UP	インベーダーを超えたゲーム?	¥ 2,000	
SLOT GAME	スロット・マシン	¥ 2,000	
CAT & RAT	追いかけっこ	¥ 2,000	
CALC	加減演算練習プログラム	¥ 2,000	
		¥	
BROWNIAN MOV	ブラウン分子運動のシュミレーション	¥ 2,000	
FROGS JUMP	カエルの入かえゲーム	¥ 3,000	
GOMOKUNARABE	五目並べ	¥ 2,000	PET
STAR TREK #2	本格的スタートレック	¥ 2,000	
BOKI	会計処理日計表等	¥ 5,000	(共立オリジナル) PET
MODULE1 ADVANCED DRAGON MAZE	迷路ゲーム	¥ 3,000	
DIGITAL DERBY	競場ゲーム		
SAUCER WAR	UFO戦争ゲーム		
MODULE2 3-D PLOT	ベッセル関数プロット	¥ 3,000	
ETCH-A-SKETCH	パドルを使って絵をかこう		
STAR	HGRで放射状にプロットします		
MODULE3 OTHELLO	オセロゲーム	¥ 3,000	
MASTERMIND	色当てゲーム		
SEVEN	7カードゲーム		
MODULE4 APPLE ODIAN	作曲、演奏、音楽プログラム	¥ 3,000	
MODULE5 AWARI	豆とりゲーム	¥ 3,000	
TOWERS OF HANOI	ハノイの塔		
HEX PAWN	16進変換プログラム		
MODULE6 BLACKJACK	トランプゲーム	¥ 3,000	
MODULE7 HYPER-LIFE	高速ライフゲーム	¥ 3,000	
8 GRAPHIC DEMOS	8種のデモパターン		
MODULE8 PROJECT UFO	4種のゲーム	¥ 3,000	APPL II
CLEAN SWEEP			
CHECK BOOK	在庫管理、小切手帳	¥10,000	
MUSIC	ドレミで作曲、演奏できる	¥ 3,000	
BULLS AND BEARS	株式と経営ゲーム	¥ 3,000	
WARLORDS	領土合戦	¥ 3,000	
MICROTRIVIA	スターや映画の名前当て	¥ 3,000	
KIDSTUFF	遊びながら英語が学べる	¥ 3,000	
FINANCIAL ANALYSIS	財務分析教育プログラム	¥ 4,800	
APPLETALKER	アップルがしゃべる!	¥ 4,800	
APPLE-LIS'NER	音声入力を認識する!	¥ 4,800	
TALKING CALCULATOR	しゃべりながら計算する	¥ 3,000	
BOMBER!	高速戦車爆撃ゲーム	¥ 3,000	
MUSIC KALEIDOSCOPE	音楽に合わせて色が踊る	¥ 3,000	
ELECTRONIC INDEX CARD FIIE	ディスク用データファイル	¥ 6,400	
APPLE-FORTH	第四世代の言語!	¥15,000	
HIRES PLOTTER	高分解能スケッチ(パドル使用)	¥ 3,000	

■市販主要ソフトテープ

ソフト名	内容	価格	備考
HIRES TEXT	高分解能テキストルーチン	¥ 3,000	
3-D HIRES GRAPHICS	立体図型形ルーチン	¥ 3,000	
CO-RESIDEMT ASSEMBLER/EDITOR	ラベルの付くアセンブラ	¥10,000	
GAME PACK(5Games)	ハングマン・ドラゴンメイズ・バイオリズム・カラーマス・カラースケッチ	¥ 4,800	
Apple RAM Test	APPLEのメモリー(RAM)テスト用プログラム	¥ 3,000	
BIORHYTHM	バイオリズム	¥ 3,000	
MICROCHESS	楽しいチェス	¥ 4,800	
U-DRAW	高分解能画面エディタ	¥ 6,400	
SHAPE GENERATOR/MARGER	図形の作成ルーチン	¥ 6,500	
PIERO	風船割りに命をかける!	¥ 3,000	
FORTE	使って楽しい音楽言語	¥ 4,800	
GOLF	君の部屋でもゴルフができる	¥ 6,400	
JUPITER EXPRESS	宇宙船操縦ゲーム	¥ 3,000	
マスターマインド/バイオリズム	色あてゲーム/バイオリズム	¥ 2,400	
ハッスル/ピンボール	パドル使用・ヘビによるカエル取り/ピンボール	¥ 2,400	
ショットアウト	パドル使用、決闘ゲーム	¥ 2,200	
オセロ	パドル使用、2人用オセロ	¥ 2,600	
スロットマシン/ブラックジャック		¥ 2,400	
シンクザシップ/ハノイの塔		¥ 2,400	
アップル21	ハイリゾ使用ブラックジャック	¥ 4,000	
3Dドッキングミッション	平面図、側面図より障害物をさけるゲーム	¥ 4,000	
ディープチャージ	潜水艦爆破ゲーム	¥ 4,000	APPL II
ザップ	パドル使用撃墜ゲーム	¥ 4,000	
ルーレット/スロットマシン(APPLESOFT II)		¥ 4,000	
ブラックジャック/クラップス(APPLESOFT II)	ブラックジャック/サイコロゲーム	¥ 4,000	
ツーボイスミュージック/バッハ&グラフィック	音楽の二重演奏	¥ 3,000	
ミュージック&グラフィック	音楽演奏	¥ 3,000	
カレイドスコープ	万華鏡	¥ 3,000	
エンジン	エンジンのシュミレーション	¥ 3,000	
数学 数列シリーズ レッスン1・2		¥ 2,300	
数学 数列シリーズ レッスン3・4		¥ 2,300	
数学 数列シリーズ レッスン5・6		¥ 2,300	
数学 数列シリーズ レッスン7・8・クイズ		¥ 2,300	
英単語シリーズ レッスン1・2・3		¥ 2,500	
英単語シリーズ レッスン4・5・6		¥ 2,500	
英単語シリーズ レッスン7・8・9		¥ 2,500	
英単語シリーズ レッスン10・クイズ		¥ 2,500	
カラーマス/ハングマン	計算問題/数あてゲーム	¥ 2,400	
モールス信号		¥ 2,400	
ロンゲビィティ(APPLESOFT II)	あなたは何才まで生きられるか?	¥ 4,500	
ウエイトコントロール(APPLESOFT II)		¥ 4,500	
バイオリズム(APPLESOFT II)		¥ 4,500	
INVADERS	パドル使用本格的インベーダーゲーム	¥ 5,800	
サカナツリ	(RAM 24K以上)	¥ 2,000	MZ80K
マスターマインド	数あてゲーム	¥ 2,000	(富士音響
高速ライフゲーム		¥ 2,600	オリジナル)
12Kスタートレック	(富士音響オリジナル)	¥ 3,000	MB6880 L2
ADO	ゲーム(文字、図案作成プログラム)	¥ 3,000	用
LANDING	着艦ゲーム	¥ 2,500	
EE-1	英語学習プログラム	¥ 2,500	
STC	業務用ソフト在庫管理	¥20,000	
MNG	諸表管理	¥10,000	
CST	顧客管理	¥20,000	
インベーダーゲーム	(共立オリジナル)	¥ 3,500	二つが入ったもの ¥4,500
ブロッククズシ	(共立オリジナル)	¥ 2,500	
7Kスタートレック	TK80BS-L2用	¥ 3,000	TK80BS L2
〃	COMPO BS/80用	¥ 3,000	(富士音響)

SHARP

パーソナルコンピュータ

MZ-80Kは、世界の最先端をいく8ビットマイコンZ-80の機能を最大限にいかしたパーソナルコンピュータの傑作です。
使用言語は、高級言語「BASIC」。入門者でも、手軽にプログラムが作成できます。しかも、ソフト、ハード両面で柔軟に拡張できる「クリーンコンピュータ」ですから、幅広く専門分野での利用も可能です。

- 言語の進化への対応や、他の言語への変更を容易にするため、内部記憶回路の固定化(ROM=Read only memory)を最少限にとどめ、フリーメモリ(RAM=Random access memory)を多く利用しています。
- 別売の拡張システムを使ってさらに多彩な発展ができるよう、バスラインを外部端子(I/Oターミナル)にまとめています。

▲MZ-80Kの主な特長

- BASIC(テープモード)
- 市販のカセットテープにプログラムの記録保存ができ、プログラムファイル名で呼出し可能。
- カセットの記憶方式はパルス幅変調方式でスピードは1200bit/秒。
- 英字、カナ文字、62種の図形、13種の漢字のキャラクターを持ち、豊富な図形処理が可能。(78キーにより204種の表示可能)CRTディスプレイ(40字×25字)
- スクリーンエディット機能装備。
- 音楽の自動演奏がBASICソフト処理で可能。
- 時計回路内蔵。
- CPUボート・CRTディスプレイ・電源等、調整、検査済のセミキット。
- Z-80バスラインI/Oによる多用接続可能。

▲別売

RAMオプション／16Kバイト……標準価格44,000円
／4Kバイト……標準価格11,000円
ハイスピードベーシック／SP-5010…標準価格 3,000円
マシンランゲージ……標準価格 6,000円
専用カバー／80Z-CVR……標準価格 3,500円
アッセンブラー・エディター、ローダー・デバッカー セット 標準価格20,000円 8月末発売
放電型プリンター／MZ-80P2…標準価格148,000円
インターフェイスユニット……標準価格29,800円
ユニバーサルI/Oカード……標準価格15,000円
フロッピーディスク……発売予定
カラーディスプレイ……発売予定

MZ-80K

標準価格 198,000円(セミキット)

シャープ株式会社

本社〒545 大阪市阿倍野区長池町22番22号電話(06)621-1221(大代表)
●お問い合わせは…本社内商品信頼性本部サービス企画部
札幌(011)551-4649・仙台(0222)96-4649・栃木(0286)37-1178・東京(03)893-4649・石川(0762)49-4649・名古屋(0568)73-4649・大阪(06)643-4649・広島(08287)4-4649・香川(0878)33-4649・福岡(092)572-4649・沖縄(0988)62-2231

資料請求券
MZ-80K
マイコン・別係

電波新聞社 〒141東京都品川区東五反田1－11－15電話03(445)6111 定価1300円 雑誌69185－11

マイコンプログラム全集① 昭和五十四年十月三十日発行 電波新聞社 〒141 東京都品川区東五反田一—十一—十五 (〇三)四四五—六一一一(大代)